
NLSY79 APPENDIX 1:
EMPLOYMENT STATUS RECODE VARIABLES
1979-1998

VARIABLE CREATION: ESR 1979–1998

Employment Status Recode (ESR) is a standard measure of the respondent's main labor force activity during the survey week (Sunday through Saturday preceding the interview date). What follows is a PL/I adaptation of a Fortran program used by the Census Bureau to create ESR for 1979-1993, and for 1994 through 1998, an SPSS adaptation of the decision rules provided by the Census Bureau to create MLR using the new CAPI-generated CPS data. Note that ESR was not created in 2000 because the CPS section on activity in the week before the survey was not included in that survey.

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

/*1979-83*/ DCL 1 ESR_VARIABLES(5),
  5 ABSENT,           5 ESR,                 5 NEWJOB,
  5 ANYWK,            5 ESRC,                5 OCCTYP,
  5 DO4WKS,           5 HRSWK,               5 WEEKS,
  5 DOING,             5 INDTYP,              5 WEIGHT,
  5 EMPLOY,            5 LOOK,                 5 WHYNOT,
  5 EMPTYP,           5 MILITARY,            5 WHYOFF;

/* 1979 VARIABLES */

DOING(1)=R(434.);                                if R(434.)<1 ! R(434.)>7 then DOING(1)=na;
ANYWK(1)=R(435.);                                if R(435.)<0 ! R(435.)>1 then ANYWK(1)=na;
HRSWK(1)=R(436.);                                if R(436.)<1 ! R(436.)>96 then HRSWK(1)=na;
                                                    if R(444.)>0 then HRSWK(1)=R(444.);
                                                    if R(450.)>0 then HRSWK(1)=R(450.);
                                                    if R(437.)<1 ! R(437.)>3 then HRANGE(1)=na;
                                                    if R(438.)<0 ! R(438.)>1 then HRS35(1)=na;
                                                    if R(439.)<1 ! R(439.)>19 then REAS35(1)=na;
                                                    if R(440.)>0 then REAS35(1)=R(440.);
                                                    if R(445.)=1 then HRANGE(1)=R(445.);
                                                    if R(446.)>0 then REAS35(1)=R(446.);
                                                    if R(441.)<0 ! R(441.)>1 then OFFWORK(1)=na;
                                                    if R(447.)<0 ! R(447.)>1 then OTIME(1)=na;
                                                    if R(452.)<0 ! R(452.)>1 then ABSENT(1)=na;
                                                    if R(453.)<1 R(453.)>9 then WHYOFF(1)=na;
                                                    if R(459.)<0 ! R(459.)>3 then SALARY(1)=na;
                                                    if R(460.)<0 ! R(460.)>1 then WORK35(1)=na;
                                                    if R(511.)<0 ! R(511.)>1 then LOOK(1)=na;
OFFWORK(1)=R(441.);                               if R(517.)<1 ! R(517.)>11 then WHYLK(1)=na;
OTIME(1)=R(447.);                                if R(457.)<1 ! R(457.)>96 then WEEKS(1)=na;
ABSENT(1)=R(452.);                                if R(519.)>0 then WEEKS(1)=R(519.);
WHYOFF(1)=R(453.);                                if R(523.)>0 then WEEKS(1)=R(523.);
SALARY(1)=R(459.);                                if R(458.)<1 ! R(458.)>2 then WKTIME(1)=na;
WORK35(1)=R(460.);                                if R(520.)>0 then WKTIME(1)=R(520.);
LOOK(1)=R(511.);                                 if R(524.)>0 then WKTIME(1)=R(524.);
DO4WKS(1)=R(512.);                                if R(522.)<1 ! R(522.)>5 then WHYNOT(1)=na;
WHYLK(1)=R(517.);                                if R(521.)=0 then WHYNOT(1)=R(521.);
WEEKS(1)=R(457.);                                 if R(525.)=0 then WHYNOT(1)=R(525.);
                                                    if R(526.)>0 then WHYNOT(1)=R(526.);
                                                    if R(463.)<0 ! R(463.)>998 then INDTYP(1)=na;
WHYNOT(1)=R(522.);                                if R(468.)<0 ! R(468.)>4 then EMPTYP(1)=na;
                                                    if R(464.)<0 ! R(464.)>994 then OCCTYP(1)=na;
INDTYP(1)=R(463.);                               if R(512.)>0 ! R(513.)>0 ! R(514.)>0 ! R(515.)>0 ! R(516.)>0 then do;
EMPTYP(1)=R(468.);                               if R(512.)>1 ! R(513.)>1 ! R(514.)>1 ! R(515.)>1 ! R(516.)>1 then do;
OCCTYP(1)=R(464.);                               do I=512. to 516. ;
if R(I)=1 then R(I)=-3; end;
if DO4WKS(1)<2 then DO4WKS(1)=10; end;
if (R(512.)=1 ! R(513.)=1 ! R(514.)=1 ! R(515.)=1 ! R(516.)=1) & DO4WKS(1)^=1 then DO4WKS(1)=1;
end;
else DO4WKS(1)=na;
if INDTYP(1)^=na ! EMPTYP(1)^=na ! OCCTYP(1)^=na
  then EMPLOY(1)=1;                                else EMPLOY(1)=0;
  if R(433.)=1 then MILITARY(1)=1;                  else MILITARY(1)=na;
  WEIGHT(1)=R(2161.);
```

```

/* 1980 VARIABLES */

DOING(2)=R(2604.);                                if R(2604.)<1 ! R(2604.)>7 then DOING(2)=na;
ANYWK(2)=R(2605.);                                if R(2605.)<0 ! R(2605.)>1 then ANYWK(2)=na;
HRSWK(2)=R(2606.);                                if R(2606.)<1 ! R(2606.)>96 then HRSWK(2)=na;
                                                       if R(2614.)>0 then HRSWK(2)=R(2614.);
                                                       if R(2620.)>0 then HRSWK(2)=R(2640.);
                                                       if R(2607.)<1 ! R(2607.)>3 then HRANGE(2)=na;
H RANGE(2)=R(2607.);                                if R(2608.)<0 ! R(2608.)>1 then HRS35(2)=na;
HRS35(2)=R(2608.);                                if R(2609.)<1 ! R(2609.)>19 then REAS35(2)=na;
REAS35(2)=R(2609.);                                if R(2610.)>0 then REAS(2)=R(2610.);
                                                       if R(2615.)=1 then HRANGE(2)=R(2615.);
                                                       if R(2616.)>0 then REAS35(2)=R(2616.);
                                                       if R(2611.)<0 ! R(2611.)>1 then OFFWRK(2)=na;
OFFWRK(2)=R(2611.);                                if R(2617.)<0 ! R(2617.)>1 then OTIME(2)=na;
OTIME(2)=R(2617.);                                if R(2622.)<0 ! R(2622.)>1 then ABSENT(2)=na;
ABSENT(2)=R(2622.);                                if R(2623.)<1 ! R(2623.)>9 then WHYOFF(2)=na;
WHYOFF(2)=R(2623.);                                if R(2629.)<0 ! R(2629.)>3 then SALARY(2)=na;
SALARY(2)=R(2629.);                                if R(2630.)<0 ! R(2630.)>1 then WORK35(2)=na;
WORK35(2)=R(2630.);                                if R(2681.)<0 ! R(2681.)>1 then LOOK(2)=na;
LOOK(2)=R(2681.);                                DO4WKS(2)=na;
                                                       if R(2682.)=1 then DO4WKS(2)=1;           else if R(2683.)>0 then DO4WKS(2)=2;
                                                       else if R(2684.)>0 then DO4WKS(2)=3;   else if R(2685.)>0 then DO4WKS(2)=4;
                                                       else if R(2686.)>0 then DO4WKS(2)=5;   else if R(2687.)>0 then DO4WKS(2)=6;
                                                       else if R(2688.)>0 then DO4WKS(2)=7;   else if R(2689.)>0 then DO4WKS(2)=8;
                                                       else if R(2690.)>0 then DO4WKS(2)=9;
WHYLK(2)=R(2691.);                                if R(2691.)<1 ! R(2691.)>11 then WHYLK(2)=na;
WEEKS(2)=R(2627.);                                if R(2627.)<1 ! R(2627.)>96 then WEEKS(2)=na;
                                                       if R(2693.)>0 then WEEKS(2)=R(2693.);
                                                       if R(2697.)>0 then WEEKS(2)=R(2697.);
WKTIME(2)=R(2628.);                                if R(2628.)<1 ! R(2628.)>2 then WKTIME(2)=na;
                                                       if R(2694.)>0 then WKTIME(2)=R(2694.); if R(2698.)>0 then WKTIME(2)=R(2698.);
                                                       if R(2696.)<1 ! R(2696.)>5 then WHYNOT(2)=na;
WHYNOT(2)=R(2696.);                                if R(2695.)=0 then WHYNOT(2)=R(2695.); if R(2699.)=0 then WHYNOT(2)=R(2699.); if R(2700.)>0 then WHYNOT(2)=R(2700.); if R(2633.)<0 ! R(2633.)>998 then INDTYP(2)=na;
INDTYP(2)=R(2633.);                                if R(2635.)<0 ! R(2635.)>4 then EMPTYYP(2)=na;
EMPTYYP(2)=R(2635.);                                OCCTYP(2)=R(2634.); if R(2634.)<0 ! R(2634.)>994 then OCCTYP(2)=na;
                                                       if R(2682.)>0 ! R(2683.)>0 ! R(2684.)>0 ! R(2685.)>0 ! R(2686.)>0 ! R(2687.)>0 ! R(2688.)>0 ! R(2689.)>0 !
                                                       R(2690.)>0 then do;
                                                       if R(2682.)>1 ! R(2683.)>1 ! R(2684.)>1 ! R(2685.)>1 ! R(2686.)>1 ! R(2687.)>1 ! R(2688.)>1 !
                                                       R(2689.)>1 ! R(2690.)>1 then do;
                                                       if R(2682.)=1 then R(2682.)=-3;
                                                       if DO4WKS(2)<2 then DO4WKS(2)=10;
                                                       end;
                                                       if R(2682.)=1 & DO4WKS(2)^=1 then DO4WKS(2)=1;
                                                       end;
                                                       else DO4WKS(2)=na;
                                                       if INDTYP(2)^=na ! EMPTYYP(2)^=na ! OCCTYP(2)^=na
                                                       then EMPLOY(2)=1;           else EMPLOY(2)=0;
                                                       if R(2624.)<1 ! R(2624.)>2 then NEWJOB(2)=na; else NEWJOB(2)=R(2624.); if R(2603.)=1 then MILITARY(2)=1;           else MILITARY(2)=na;

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

/ 1981 VARIABLES */*

```

if R(4434.)<1 then DOING(3)=na;
if R(4435.)<0 then ANYWK(3)=na;
if R(4436.)<1 then HRSWK(3)=na;
if R(4444.)>0 then HRSWK(3)=R(4444.);
if R(4450.)>0 then HRSWK(3)=R(4450.);
if R(4452.)<0 then ABSENT(3)=na;
if R(4453.)<1 then WHYOFF(3)=na;
if R(4495.)<0 then LOOK(3)=na;
DO4WKS(3)=na;
if R(4496.)=1 then DO4WKS(3)=1;
if R(4497.)>0 ! R(4498.)>0 ! R(4499.)>0 ! R(4500.)>0 ! R(4501.)>0 ! R(4502.)>0 ! R(4503.)>0 ! R(4504.)>0
    then DO4WKS(3)=2;
if R(4457.)<1 then WEEKS(3)=na;
if R(4507.)>0 then WEEKS(3)=R(4507.);
if R(4511.)>0 then WEEKS(3)=R(4511.);
if R(4510.)<1 then WHYNOT(3)=na;
if R(4509.)=0 ! R(4513.)=0 then WHYNOT(3)=0;
if R(4514.)>0 then WHYNOT(3)=R(4514.);
if R(4463.)<0 ! R(4463.)>998 then INDTYP(3)=na;
if R(4464.)<0 ! R(4464.)>994 then OCCTYP(3)=na;
if R(4466.)<0 then EMPTYP(3)=na;
if R(4454.)<1 ! R(4454.)>2 then NEWJOB(3)=na;
if INDTYP(3)^=na ! OCCTYP(3)^=na ! EMPTYP(3)^=na
    then EMPLOY(3)=1;
WEIGHT(3)=R(6146.);
if R(4433.)=1 then MILITARY(3)=1;

```

/ 1982 VARIABLES */*

```

if R(6935.)<1 then DOING(4)=na;
if R(6936.)<0 then ANYWK(4)=na;
if R(6937.)<1 then HRSWK(4)=na;
if R(6945.)>0 then HRSWK(4)=R(6945.);
if R(6951.)>0 then HRSWK(4)=R(6951.);
if R(6953.)<0 then ABSENT(4)=na;
if R(6954.)<1 then WHYOFF(4)=na;
if R(6963.)<0 then LOOK(4)=na;
DO4WKS(4)=na;
if R(6964.)=1 then DO4WKS(4)=1;
if R(6965.)>0 ! R(6966.)>0 ! R(6967.)>0 ! R(6968.)>0 ! R(6969.)>0 !
    R(6970.)>0 ! R(6971.)>0 ! R(6972.)>0 then DO4WKS(4)=2;
if R(6958.)<1 then WEEKS(4)=na;
if R(6975.)>0 then WEEKS(4)=R(6975.);
if R(6979.)>0 then WEEKS(4)=R(6979.);
if R(6978.)<1 then WHYNOT(4)=na;
if R(6977.)=0 ! R(6981.)=0 then WHYNOT(4)=0;
if R(6982.)>0 then WHYNOT(4)=R(6982.);
if R(7020.)<0 ! R(7020.)>998 then INDTYP(4)=na;
if R(7021.)<0 ! R(7021.)>994 then OCCTYP(4)=na;
if R(7023.)<0 then EMPTYP(4)=na;
if R(6955.)<1 ! R(6955.)>2 then NEWJOB(4)=na;
if (INDTYP(4)^=na ! OCCTYP(4)^=na ! EMPTYP(4)^=na) & R(6983.)^=1 & R(7009.)^=1
    then EMPLOY(4)=1;
WEIGHT(4)=R(8967.);
if R(6934.)=1 then MILITARY(4)=1;

```

```

else DOING(3)=R(4434.);
else ANYWK(3)=R(4435.);
else HRSWK(3)=R(4436.);

else ABSENT(3)=R(4452.);
else WHYOFF(3)=R(4453.);
else LOOK(3)=R(4495.);

else WEEKS(3)=R(4457.);

else WHYNOT(3)=R(4510.);

else INDTYP(3)=R(4463.);
else OCCTYP(3)=R(4464.);
else EMPTYP(3)=R(4466.);
else NEWJOB(3)=R(4454.);

else EMPLOY(3)=0;

else MILITARY(3)=na;

```

```

else DOING(4)=R(6935.);
else ANYWK(4)=R(6936.);
else HRSWK(4)=R(6937.);

else ABSENT(4)=R(6953.);
else WHYOFF(4)=R(6954.);
else LOOK(4)=R(6963.);

else WEEKS(4)=R(6958.);

else WHYNOT(4)=R(6978.);

else INDTYP(4)=R(7020.);
else OCCTYP(4)=R(7021.);
else EMPTYP(4)=R(7023.);
else NEWJOB(4)=R(6955.);

else EMPLOY(4)=0;

else MILITARY(4)=na;

```

/* 1983 VARIABLES */

```

if R(9365.)<1 then DOING(5)=na;
if R(9366.)<0 then ANYWK(5)=na;
if R(9367.)<1 then HRSWK(5)=na;
if R(9375.)>0 then HRSWK(5)=R(9375.);
if R(9381.)>0 then HRSWK(5)=R(9381.);
if R(9383.)<0 then ABSENT(5)=na;
if R(9384.)<1 then WHYOFF(5)=na;
if R(9393.)<0 then LOOK(5)=na;
DO4WKS(5)=na;
if R(9394.)=1 then DO4WKS(5)=1;
if R(9395.)>0 ! R(9396.)>0 ! R(9397.)>0 ! R(9398.)>0 ! R(9399.)>0 !
    R(9400.)>0 ! R(9401.)>0 ! R(9402.)>0 then DO4WKS(5)=2;
if R(9388.)<1 then WEEKS(5)=na;
if R(9405.)>0 then WEEKS(5)=R(9405.);
if R(9408.)>0 then WEEKS(5)=R(9408.);
if R(9407.)=0 ! R(9410.)=0 then WHYNOT(5)=0;
if R(9411.)>0 then WHYNOT(5)=R(9411.);
if R(9449.)<0 ! R(9449.)>998 then INDTYP(5)=na;
if R(9450.)<0 ! R(9450.)>994 then OCCTYP(5)=na;
if R(9454.)<0 then EMPTYP(5)=na;
if R(9385.)<1 ! R(9385.)>2 then NEWJOB(5)=na;
if (INDTYP(5)^=na ! OCCTYP(5)^=na ! EMPTYP(5)^=na) & R(9412.)^=1 & R(9439.)^=1
    then EMPLOY(5)=1;
WEIGHT(5)=R(11444.);
if R(9364.)=1 then MILITARY(5)=1;
do I=1 to 5;
    if MILITARY(I)=1 then do; ESR(I)=8; go to fin; end;
    if I=1 & R(1.)=6435 then do; ESR(1)=-3; go to fin; end;
    if WEIGHT(I)=0 then do; ESR(I)=-5; go to fin; end;
    ESR(I)=7;
    if DOING(I)^=1 then go to P6;
    if HRSWK(I)=na then go to P4;
    if HRSWK(I)>=15 ! EMPTYP(I)^=4 then ESR(I)=1;
P4: if EMPLOY(I)=1 then ESR(I)=1;
P6: ESR(I)=1;
    if ANYWK(I)^=1 then go to P11;
    if HRSWK(I)=na then go to P9;
    if HRSWK(I)<15 & EMPTYP(I)=4 then go to P61;
P9: if EMPLOY(I)=0 then go to P61;
P11: if HRSWK(I)=na then go to P15;
    if EMPLOY(I)=0 then go to P61;
    if HRSWK(I)<15 & EMPTYP(I)=4 then go to P61;
P15: ESR(I)=7;
    if DOING(I)=2 then go to ESR237;
    if DOING(I)=3 then go to P21;
P21: if ABSENT(I)^=1 then go to P27;
    P27: if WHYOFF(I)=na then go to P32;
    if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P29;
    if EMPLOY(I)^=0 & EMPTYP(I)^=4 then ESR(I)=2;
P29: if WHYNOT(I)<3 then ESR(I)=3;
P32: if DO4WKS(I)=1 then go to P20000;
    if WHYNOT(I)>2 then go to P20000;
    if WEEKS(I)=na then go to P33;
    ESR(I)=3;
P33: if DO4WKS(I)>1 ! WHYNOT(I)^=na then ESR(I)=3;
P37: ESR(I)=2;

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

if ABSENT(I)^=1 then go to P43;
if WHYOFF(I)=na then go to P41;
if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P39;
if EMPTYP(I)=4 then go to P61;                                go to P62;
P39: if WHYNOT(I)>2 then go to P61;
      ESR(I)=3;                                              go to P62;
P41: if EMPLOY(I)=0 ! EMPTYP(I)=4 then go to P61;            go to P62;
P43: if WHYOFF(I)=na then go to P48;                          go to P62;
if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P45;
if EMPLOY(I)=0 ! EMPTYP(I)=4 then go to P61;                  go to P62;
P45: if WEEKS(I)=na & WHYNOT(I)=na then go to P61;
if WHYNOT(I)>2 then go to P61;
      ESR(I)=3;                                              go to P62;
P48: ESR(I)=3;
if LOOK(I)^=1 then go to P53;
if DO4WKS(I)=1 then go to P61;
if WHYNOT(I)>2 then go to P61;
if DO4WKS(I)^=na ! WEEKS(I)^=na then go to P62;
if WHYNOT(I)=na then go to P61;                                go to P62;
P53: if DO4WKS(I)=1 then go to P61;
if WHYNOT(I)>2 then go to P61;
if DO4WKS(I)=na then go to P56;
if WEEKS(I)^=na ! WHYNOT(I)^=na then go to P62;                go to P61;
P56: if WEEKS(I)=na ! WHYNOT(I)=na then go to P61;            go to P62;
P61: if DOING(I)<4 ! DOING(I)>6 then go to P610;
      ESR(I)=DOING(I);                                      go to P20000;
      P610: ESR(I)=7;                                     go to P20000;
P62: if DOING(I)=6 then DOING(I)=7;
P20000: go to fin;
ESR237: if WHYOFF(I)=na then go to P5019;
if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P5017;
if EMPTYP(I)^=-4 then ESR(I)=2;                                go to fin;
P5017: if WHYNOT(I)>2 then go to fin;
      ESR(I)=3;                                              go to fin;
P5019: if EMPLOY(I)=1 & EMPTYP(I)^=4 then ESR(I)=2;
fin;;
/* COLLAPSED EMPLOYMENT STATUS RECODE */

      if ESR(I)^=-5 then ESRC(I)^=-5;
      else if ESR(I)=1 | ESR(I)=2 then ESRC(I)=1;
      else if ESR(I)=3 then ESRC(I)=2;
      else if ESR(I)>=4 & ESR(I)<=7 then ESRC(I)=3;
      else if ESR(I)=8 then ESRC(I)=4;
      else ESRC(I)=ESR(I);
end;

R(2149.)=ESR(1);          R(4063.)=ESR(2);          R(6188.)=ESR(3);
R(8985.)=ESR(4);          R(11447.)=ESR(5);          R(2149.01)=ESRC(1);
R(4063.01)=ESRC(2);       R(6188.01)=ESRC(3);        R(8985.01)=ESRC(4);
R(11447.01)=ESRC(5);

```

/*1984-93*/ DCL 1 ESR (6:14),

5 ABSENT,	5 INDTYP,
5 ANYWK,	5 LOOK,
5 DO4WKS,	5 MILITARY,
5 DOING,	5 NEWJOB,
5 EMPLOY,	5 OCCTYP,
5 EMPTYP,	5 WEEKS,
5 ESR,	5 WEIGHT,
5 ESRC,	5 WHYNOT,
5 HRSWK,	5 WHYOFF;

/* 1984 VARIABLES */

```

ESR84=-4;
if R(12468.)<1 then DOING(6)=na;
if R(12469.)<0 then ANYWK(6)=na;
if R(12470.)<1 then HRSWK(6)=na;
if R(12478.)>0 then HRSWK(6)=R(12478.);
if R(12484.)>0 then HRSWK(6)=R(12484.);
if R(12486.)<0 then ABSENT(6)=na;
if R(12487.)<1 then WHYOFF(6)=na;
if R(12496.)<0 then LOOK(6)=na;
DO4WKS(6)=na;
if R(12497.)=1 then DO4WKS(6)=1;
if R(12498.)>0 ! R(12499.)>0 ! R(12500.)>0 ! R(12501.)>0 ! R(12502.)>0 ! R(12503.)>0 ! R(12504.)>0 !
R(12505.)>0 then DO4WKS(6)=2;
if R(12491.)<1 then WEEKS(6)=na;
if R(12508.)>0 then WEEKS(6)=R(12508.);
if R(12512.)>0 then WEEKS(6)=R(12512.);
if R(12510.)=0 ! R(12514.)=0 then WHYNOT(6)=0;
if R(12515.)>0 then WHYNOT(6)=R(12515.);
if R(12553.)<0 ! R(12553.)>998 then INDTYP(6)=na;
if R(12554.)<0 ! R(12554.)>994 then OCCTYP(6)=na;
if R(12558.)<0 then EMPTYP(6)=na;
if R(12488.)<1 ! R(12488.)>2 then NEWJOB(6)=na;
if (INDTYP(6)^=na ! OCCTYP(6)^=na ! EMPTYP(6)^=na) & R(12516.)^=1 & R(12543.)^=1
then EMPLOY(6)=1;
WEIGHT(6)=R(15196.);
if R(12467.)=1 then MILITARY(6)=1;
else INDTYP(6)=R(12553.);
else OCCTYP(6)=R(12554.);
else EMPTYP(6)=R(12558.);
else NEWJOB(6)=R(12488.);
else EMPLOY(6)=0;
else MILITARY(6)=na;

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

/ 1985 VARIABLES */*

```

if R(16420.)<1 then DOING(7)=na;
if R(16421.)<0 then ANYWK(7)=na;
if R(16422.)<1 then HRSWK(7)=na;
if R(16430.)>0 then HRSWK(7)=R(16430.);
if R(16436.)>0 then HRSWK(7)=R(16436.);
if R(16438.)<0 then ABSENT(7)=na;
if R(16439.)<1 then WHYOFF(7)=na;
if R(16448.)<0 then LOOK(7)=na;
DO4WKS(7)=na;
if R(16449.)=1 then DO4WKS(7)=1;
if R(16450.)>0 ! R(16451.)>0 ! R(16452.)>0 ! R(16453.)>0 ! R(16454.)>0 ! R(16455.)>0 ! R(16456.)>0 !
    R(16457.)>0 then DO4WKS(7)=2;
if R(16443.)<1 then WEEKS(7)=na;
if R(16460.)>0 then WEEKS(7)=R(16460.);
if R(16464.)>0 then WEEKS(7)=R(16464.);
if R(16462.)=0 ! R(16466.)=0 then WHYNOT(7)=0;
if R(16467.)>0 then WHYNOT(7)=R(16467.);
if R(16501.)<0 ! R(16501.)>998 then INDTYP(7)=na;
if R(16502.)<0 ! R(16502.)>994 then OCCTYP(7)=na;
if R(16506.)<0 then EMPTYP(7)=na;
if R(16440.)<1 ! R(16440.)>2 then NEWJOB(7)=na;
if (INDTYP(7)^=na ! OCCTYP(7)^=na ! EMPTYP(7)^=na) & R(16468.)^=1 & R(16493.)^=1
    then EMPLOY(7)=1;
WEIGHT(7)=R(18902.);
if R(16419.)=1 then MILITARY(7)=1;

else DOING(7)=R(16420.);
else ANYWK(7)=R(16421.);
else HRSWK(7)=R(16422.);

else ABSENT(7)=R(16438.);
else WHYOFF(7)=R(16439.);
else LOOK(7)=R(16448.);

else WEEKS(7)=R(16443.);

else INDTYP(7)=R(16501.);
else OCCTYP(7)=R(16502.);
else EMPTYP(7)=R(16506.);
else NEWJOB(7)=R(16440.);

else EMPLOY(7)=0;

else MILITARY(7)=na;

```

/ 1986 VARIABLES */*

```

if R(19146.)<1 then DOING(8)=na;
if R(19147.)<0 then ANYWK(8)=na;
if R(19148.)<1 then HRSWK(8)=na;
if R(19156.)>0 then HRSWK(8)=R(19156.);
if R(19162.)>0 then HRSWK(8)=R(19162.);
if R(19164.)<0 then ABSENT(8)=na;
if R(19165.)<1 then WHYOFF(8)=na;
if R(19174.)<0 then LOOK(8)=na;
DO4WKS(8)=na;
if R(19175.)=1 then DO4WKS(8)=1;
if R(19176.)>0 ! R(19177.)>0 ! R(19178.)>0 ! R(19179.)>0 ! R(19180.)>0 ! R(19181.)>0 ! R(19182.)>0 !
    R(19183.)>0 then DO4WKS(8)=2;
if R(19169.)<1 then WEEKS(8)=na;
if R(19186.)>0 then WEEKS(8)=R(19186.);
if R(19190.)>0 then WEEKS(8)=R(19190.);
if R(19188.)=0 ! R(19192.)=0 then WHYNOT(8)=0;
if R(19193.)>0 then WHYNOT(8)=R(19193.);
if R(19227.)<0 ! R(19227.)>998 then INDTYP(8)=na;
if R(19228.)<0 ! R(19228.)>994 then OCCTYP(8)=na;
if R(19232.)<0 then EMPTYP(8)=na;
if R(19166.)<1 ! R(19166.)>2 then NEWJOB(8)=na;
if (INDTYP(8)^=na ! OCCTYP(8)^=na ! EMPTYP(8)^=na) & R(19194.)^=1 & R(19219.)^=1
    then EMPLOY(8)=1;
WEIGHT(8)=R(22573.);
MILCUR=R(19115.);
if R(19141.)>=0 then MILCUR=R(19141.); if MILCUR=1 then MILITARY(8)=1; else MILITARY(8)=na;

else DOING(8)=R(19146.);
else ANYWK(8)=R(19147.);
else HRSWK(8)=R(19148.);

else ABSENT(8)=R(19164.);
else WHYOFF(8)=R(19165.);
else LOOK(8)=R(19174.);

else WEEKS(8)=R(19169.);

else INDTYP(8)=R(19227.);
else OCCTYP(8)=R(19228.);
else EMPTYP(8)=R(19232.);
else NEWJOB(8)=R(19166.);

else EMPLOY(8)=0;

else MILITARY(8)=na;

```

/* 1987 VARIABLES */

```

if R(23107.)<1 then DOING(9)=na;
if R(23108.)<0 then ANYWK(9)=na;
if R(23109.)<1 then HRSWK(9)=na;
if R(23117.)>0 then HRSWK(9)=R(23117.);
if R(23123.)>0 then HRSWK(9)=R(23123.);
if R(23125.)<0 then ABSENT(9)=na;
if R(23126.)<1 then WHYOFF(9)=na;
if R(23135.)<0 then LOOK(9)=na;
DO4WKS(9)=na;
if R(23137.)>0 ! R(23138.)>0 ! R(23139.)>0 ! R(23140.)>0 ! R(23141.)>0 ! R(23142.)>0 ! R(23143.)>0 !
    R(23144.)>0 then DO4WKS(9)=2;
if R(23130.)<1 then WEEKS(9)=na;
if R(23147.)>0 then WEEKS(9)=R(23147.);
if R(23151.)>0 then WEEKS(9)=R(23151.);
if R(23149.)=0 ! R(23153.)=0 then WHYNOT(9)=0;
else if R(23154.)>0 then WHYNOT(9)=R(23154.);
else if R(23150.)>0 then WHYNOT(9)=R(23150.);
if R(23175.)<0 ! R(23175.)>998 then INDTYP(9)=na;
if R(23176.)<0 ! R(23176.)>994 then OCCTYP(9)=na;
if R(23180.)<0 then EMPTYP(9)=na;
if R(23127.)<1 ! R(23127.)>2 then NEWJOB(9)=na;
if (INDTYP(9)^=na ! OCCTYP(9)^=na ! EMPTYP(9)^=na) & R(23155.)^=1 & R(23174.)^=1
    then EMPLOY(9)=1;
    WEIGHT(9)=R(24445.);
    MILCUR=R(23075.);
if R(23102.)>=0 then MILCUR=R(23102.); if MILCUR=1 then MILITARY(9)=1; else MILITARY(9)=na;
else DOING(9)=R(23107.);
else ANYWK(9)=R(23108.);
else HRSWK(9)=R(23109.);

else ABSENT(9)=R(23125.);
else WHYOFF(9)=R(23126.);
else LOOK(9)=R(23135.);
if R(23136.)=1 then DO4WKS(9)=1;
else WEEKS(9)=R(23130.);

else WHYNOT(9)=na;
else INDTYP(9)=R(23175.);
else OCCTYP(9)=R(23176.);
else EMPTYP(9)=R(23180.);
else NEWJOB(9)=R(23127.);
else EMPLOY(9)=0;

```

/* 1988 VARIABLES */

```

if R(25186.)<1 then DOING(10)=na;
if R(25187.)<0 then ANYWK(10)=na;
if R(25188.)<1 then HRSWK(10)=na;
if R(25196.)>0 then HRSWK(10)=R(25196.);
if R(25202.)>0 then HRSWK(10)=R(25202.);
if R(25204.)<0 then ABSENT(10)=na;
if R(25205.)<1 then WHYOFF(10)=na;
if R(25214.)<0 then LOOK(10)=na;
DO4WKS(10)=na;
if R(25216.)>0 ! R(25217.)>0 ! R(25218.)>0 ! R(25219.)>0 ! R(25220.)>0 ! R(25221.)>0 ! R(25222.)A>0 !
    R(25223.)>0 then DO4WKS(10)=2;
if R(25209.)<1 then WEEKS(10)=na;
if R(25226.)>0 then WEEKS(10)=R(25226.);
if R(25230.)>0 then WEEKS(10)=R(25230.);
if R(25228.)=0 ! R(25232.)=0 then WHYNOT(10)=0;
else if R(25233.)>0 then WHYNOT(10)=R(25233.);
else if R(25229.)>0 then WHYNOT(10)=R(25229.);
if R(25253.)<0 ! R(25253.)>998 then INDTYP(10)=na;
if R(25254.)<0 ! R(25254.)>994 then OCCTYP(10)=na;
if R(25258.)<0 then EMPTYP(10)=na;
if R(25206.)<1 ! R(25206.)>2 then NEWJOB(10)=na;
if (INDTYP(10)^=na ! OCCTYP(10)^=na ! EMPTYP(10)^=na) & R(25234.)^=1 & R(25252.)^=1
    then EMPLOY(10)=1;
    WEIGHT(10)=R(28700.);
    MILCUR=R(25154.);
if R(25181.)>=0 then MILCUR=R(25181.); if MILCUR=1 then MILITARY(10)=1; else MILITARY(10)=na;
else DOING(10)=R(25186.);
else ANYWK(10)=R(25187.);
else HRSWK(10)=R(25188.);

else ABSENT(10)=R(25204.);
else WHYOFF(10)=R(25205.);
else LOOK(10)=R(25214.);
if R(25215.)=1 then DO4WKS(10)=1;
else WEEKS(10)=R(25209.);

else WHYNOT(10)=na;
else INDTYP(10)=R(25253.);
else OCCTYP(10)=R(25254.);
else EMPTYP(10)=R(25258.);
else NEWJOB(10)=R(25206.);
else EMPLOY(10)=0;

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

/ 1989 VARIABLES */*

```

if R(29175.)<1 then DOING(11)=na;
if R(29176.)<0 then ANYWK(11)=na;
if R(29177.)<1 then HRSWK(11)=na;
if R(29185.)>0 then HRSWK(11)=R(29185.);
if R(29191.)>0 then HRSWK(11)=R(29191.);
if R(29193.)<0 then ABSENT(11)=na;
if R(29194.)<1 then WHYOFF(11)=na;
if R(29203.)<0 then LOOK(11)=na;
DO4WKS(11)=na;
if R(29205.)>0 ! R(29206.)>0 ! R(29207.)>0 ! R(29208.)>0 ! R(29209.)>0 ! R(29210.)>0 ! R(29211.)>0 !
    R(29212.)>0 then DO4WKS(11)=2;
if R(29198.)<1 then WEEKS(11)=na;
if R(29215.)>0 then WEEKS(11)=R(29215.);
if R(29219.)>0 then WEEKS(11)=R(29219.);
if R(29217.)=0 ! R(29221.)=0 then WHYNOT(11)=0;
else if R(29222.)>0 then WHYNOT(11)=R(29222.);
else if R(29218.)>0 then WHYNOT(11)=R(29218.);
if R(29243.)<0 ! R(29243.)>998 then INDTYP(11)=na;
if R(29244.)<0 ! R(29244.)>994 then OCCTYP(11)=na;
if R(29248.)<0 then EMPTYP(11)=na;
if R(29195.)<1 ! R(29195.)>2 then NEWJOB(11)=na;
if (INDTYP(11)^=na ! OCCTYP(11)^=na ! EMPTYP(11)^=na) & R(29223.)^=1 & R(29242.)^=1
    then EMPLOY(11)=1;
WEIGHT(11)=R(30738.);
MILCUR=R(29143.);
if R(29170.)>=0 then MILCUR=R(29170.); if MILCUR=1 then MILITARY(11)=1; else MILITARY(11)=na;

```

/ 1990 VARIABLES */*

```

WEIGHT(12)=R(34002.);
if R(31197.)<1 then DOING(12)=na;
if R(31198.)<0 then ANYWK(12)=na;
if R(31199.)<1 then HRSWK(12)=na;
if R(31207.)>0 then HRSWK(12)=R(31207.);
if R(31213.)>0 then HRSWK(12)=R(31213.);
if R(31220.)<0 then ABSENT(12)=na;
if R(31221.)<1 then WHYOFF(12)=na;
if R(31230.)<0 then LOOK(12)=na;
DO4WKS(12)=na;
if R(31232.)>0 ! R(31233.)>0 ! R(31234.)>0 ! R(31235.)>0 ! R(31236.)>0 ! R(31237.)>0 ! R(31238.)>0 !
    R(31239.)>0 then DO4WKS(12)=2;
if R(31225.)<1 then WEEKS(12)=na;
if R(31242.)>0 then WEEKS(12)=R(31242.);
if R(31246.)>0 then WEEKS(12)=R(31246.);
if R(31244.)=0 ! R(31248.)=0 then WHYNOT(12)=0;
else if R(31249.)>0 then WHYNOT(12)=R(31249.);
else if R(31245.)>0 then WHYNOT(12)=R(31245.);
if R(31270.)<0 ! R(31270.)>998 then INDTYP(12)=na;
if R(31271.)<0 ! R(31271.)>994 then OCCTYP(12)=na;
if R(31275.)<0 then EMPTYP(12)=na;
if R(31222.)<1 ! R(31222.)>2 then NEWJOB(12)=na;
if (INDTYP(12)^=na ! OCCTYP(12)^=na ! EMPTYP(12)^=na) & R(31250.)^=1 & R(31271.)^=1
    then EMPLOY(12)=1;
MILCUR=R(31165.);
if R(31192.)>=0 then MILCUR=R(31192.); if MILCUR=1 then MILITARY(12)=1; else MILITARY(12)=na;

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

/ 1991 VARIABLES */*

```

WEIGHT(13)=R(36558.);
if R(35154.)<1 then DOING(13)=na;
if R(35155.)<0 then ANYWK(13)=na;
if R(35156.)<1 then HRSWK(13)=na;
if R(35164.)>0 then HRSWK(13)=R(35164.);
if R(35170.)>0 then HRSWK(13)=R(35170.);
if R(35177.)<0 then ABSENT(13)=na;
if R(35178.)<1 then WHYOFF(13)=na;
if R(35187.)<0 then LOOK(13)=na;
DO4WKS(13)=na;
if R(35189.)>0 ! R(35190.)>0 ! R(35191.)>0 ! R(35192.)>0 ! R(35193.)>0 ! R(35194.)>0 ! R(35195.)>0 !
    R(35196.)>0 then DO4WKS(13)=2;
if R(35182.)<1 then WEEKS(13)=na;
if R(35199.)>0 then WEEKS(13)=R(35199.);
if R(35203.)>0 then WEEKS(13)=R(35203.);
if R(35201.)=0 ! R(35205.)=0 then WHYNOT(13)=0;
else if R(35206.)>0 then WHYNOT(13)=R(35206.);
else if R(35202.)>0 then WHYNOT(13)=R(35202.);
if R(35227.)<0 ! R(35227.)>998 then INDTYP(13)=na;
if R(35228.)<0 ! R(35228.)>994 then OCCTYP(13)=na;
if R(35232.)<0 then EMPTYP(13)=na;
if R(35179.)<1 ! R(35179.)>2 then NEWJOB(13)=na;
if (INDTYP(13)^=na ! OCCTYP(13)^=na ! EMPTYP(13)^=na) & R(35207.)^=1 & R(35228.)^=1
    then EMPLOY(13)=1;
MILCUR=R(35119.);
if R(35149.)>=0 then MILCUR=R(35149.); if MILCUR=1 then MILITARY(13)=1; else MILITARY(13)=na;

else DOING(13)=R(35154.);
else ANYWK(13)=R(35155.);
else HRSWK(13)=R(35156.);

else ABSENT(13)=R(35177.);
else WHYOFF(13)=R(35178.);
else LOOK(13)=R(35187.);
if R(35188.)=1 then DO4WKS(13)=1;
else WEEKS(13)=R(35182.);

else WHYNOT(13)=na;
else INDTYP(13)=R(35227.);
else OCCTYP(13)=R(35228.);
else EMPTYP(13)=R(35232.);
else NEWJOB(13)=R(35179.);
else EMPLOY(13)=0;

else DOING(14)=R(37203.);
else ANYWK(14)=R(37204.);
else HRSWK(14)=R(37205.);

else ABSENT(14)=R(37226.);
else WHYOFF(14)=R(37227.);
else LOOK(14)=R(37236.);
if R(37237.)=1 then DO4WKS(14)=1;
else WEEKS(14)=R(37231.);

else WHYNOT(14)=na;
else INDTYP(14)=R(37277.);
else OCCTYP(14)=R(37278.);
else EMPTYP(14)=R(37282.);
else NEWJOB(14)=R(37228.);
else EMPLOY(14)=0;

```

/ 1992 VARIABLES */*

```

WEIGHT(14)=R(40063.);
if R(37203.)<1 then DOING(14)=na;
if R(37204.)<0 then ANYWK(14)=na;
if R(37205.)<1 then HRSWK(14)=na;
if R(37213.)>0 then HRSWK(14)=R(37213.);
if R(37219.)>0 then HRSWK(14)=R(37219.);
if R(37226.)<0 then ABSENT(14)=na;
if R(37227.)<1 then WHYOFF(14)=na;
if R(37236.)<0 then LOOK(14)=na;
DO4WKS(14)=na;
if R(37238.)>0 ! R(37239.)>0 ! R(37240.)>0 ! R(37241.)>0 ! R(37242.)>0 ! R(37243.)>0 ! R(37244.)>0 !
    R(37245.)>0 then DO4WKS(14)=2;
if R(37231.)<1 then WEEKS(14)=na;
if R(37248.)>0 then WEEKS(14)=R(37248.);
if R(37252.)>0 then WEEKS(14)=R(37252.);
if R(37250.)=0 ! R(37254.)=0 then WHYNOT(14)=0;
else if R(37255.)>0 then WHYNOT(14)=R(37255.);
else if R(37251.)>0 then WHYNOT(14)=R(37251.);
if R(37277.)<0 ! R(37277.)>998 then INDTYP(14)=na;
if R(37278.)<0 ! R(37278.)>994 then OCCTYP(14)=na;
if R(37282.)<0 then EMPTYP(14)=na;
if R(37228.)<1 ! R(37228.)>2 then NEWJOB(14)=na;
if (INDTYP(14)^=na ! OCCTYP(14)^=na ! EMPTYP(14)^=na) & R(37256.)^=1 & R(37278.)^=1
    then EMPLOY(14)=1;
MILCUR=R(37168.);
if R(37198.)>=0 then MILCUR=R(37198.); if MILCUR=1 then MILITARY(14)=1; else MILITARY(14)=na;

else DOING(14)=R(37203.);
else ANYWK(14)=R(37204.);
else HRSWK(14)=R(37205.);

else ABSENT(14)=R(37226.);
else WHYOFF(14)=R(37227.);
else LOOK(14)=R(37236.);
if R(37237.)=1 then DO4WKS(14)=1;
else WEEKS(14)=R(37231.);

else WHYNOT(14)=na;
else INDTYP(14)=R(37277.);
else OCCTYP(14)=R(37278.);
else EMPTYP(14)=R(37282.);
else NEWJOB(14)=R(37228.);
else EMPLOY(14)=0;

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

/ 1993 VARIABLES */*

```

/* if KEY0002 = -4 then */
WEIGHT(15)=1; /* else WEIGHT(15)=0; */
ABSENT(15)=-4; ANYWK(15)=-4;
DOING(15)=-4; EMPLOY(15)=-4;
ESR(15)=-4; ESRC(15)=-4;
INDTYP(15)=-4; LOOK(15)=-4;
NEWJOB(15)=-4; OCCTYP(15)=-4;
WHYNOT(15)=-4; WHYOFF(15)=-4;
if R(41718.)<1 then DOING(15)=na;
if R(41719.)<0 then ANYWK(15)=na;
if R(41720.)<1 then HRSWK(15)=na;
if R(41727.)>0 then HRSWK(15)=R(41727.);
if R(41732.)>0 then HRSWK(15)=R(41732.);
if R(41741.)<0 then ABSENT(15)=na;
if R(41742.)<1 then WHYOFF(15)=na;
if R(41751.)<0 then LOOK(15)=na;
DO4WKS(15)=na;
if R(41752.)=1 then DO4WKS(15)=1;
if R(41763.)>0 then DO4WKS(15)=2;
if R(41746.)<1 then WEEKS(15)=na;
if R(41766.)>0 then WEEKS(15)=R(41766.);
if R(41770.)>0 then WEEKS(15)=R(41770.);
if R(41768.)=0 ! R(41772.)=0 then WHYNOT(15)=0;
else if R(41773.)>0 then WHYNOT(15)=R(41773.);
else if R(41769.)>0 then WHYNOT(15)=R(41769.);
if R(41820.)<0 ! R(41820.)>998 then INDTYP(15)=na;
if R(41821.)<0 ! R(41821.)>994 then OCCTYP(15)=na;
if R(41823.)<0 then EMPTYP(15)=na;
if R(41743.)<1 ! R(41743.)>2 then NEWJOB(15)=na;
if (INDTYP(15)^=na ! OCCTYP(15)^=na ! EMPTYP(15)^=na)
    then EMPLOY(15)=1;
MILCUR=R(41449.);
if R(41448.)>=0 then MILCUR=R(41448.);
if R(41460.)>=0 then MILCUR=R(41460.);
if MILCUR=1 then MILITARY(15)=1;
do I=1 to 15;
    if MILITARY(I)=1 then do; ESR(I)=8; go to fin; end;
    if WEIGHT(I)=0 then do; ESR(I)=-5; go to fin; end;
    ESR(I)=7;
    if DOING(I)^=1 then go to P6;
    if HRSWK(I)=na then go to P4;
    if HRSWK(I)>=15 ! EMPTYP(I)^=4 then ESR(I)=1;
    P4: if EMPLOY(I)=1 then ESR(I)=1;
    P6: ESR(I)=1;
    if ANYWK(I)^=1 then go to P11;
    if HRSWK(I)=na then go to P9;
    if HRSWK(I)<15 & EMPTYP(I)=4 then go to P61;
    P9: if EMPLOY(I)=0 then go to P61;
    P11: if HRSWK(I)=na then go to P15;
    if EMPLOY(I)=0 then go to P61;
    if HRSWK(I)<15 & EMPTYP(I)=4 then go to P61;
    P15: ESR(I)=7;
    if DOING(I)=2 then go to ESR237;
    if DOING(I)=3 then go to P21;
    P21: if ABSENT(I)^=1 then go to P27;
    P27: if WHYOFF(I)=na then go to P32;

```

DO4WKS(15)=-4;
 EMPTYP(15)=-4;
 HRSWK(15)=-4;
 MILITARY(15)=-4;
 WEEKS(15)=-4;

 else DOING(15)=R(41718.);
 else ANYWK(15)=R(41719.);
 else HRSWK(15)=R(41720.);

 else ABSENT(15)=R(41741.);
 else WHYOFF(15)=R(41742.);
 else LOOK(15)=R(41751.);

 else WEEKS(15)=R(41746.);

 else WHYNOT(15)=na;
 else INDTYP(15)=R(41820.);
 else OCCTYP(15)=R(41821.);
 else EMPTYP(15)=R(41823.);
 else NEWJOB(15)=R(41743.);
& R(41774.)^=1 & R(41821.)^=1
 else EMPLOY(15)=0;

 else MILITARY(15)=na;

 go to P20000;
 go to P20000;

 go to P62;
 go to P62;

 go to P62;

 go to P37;
 go to ESR237;

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P29;
if EMPLOY(I)^=0 & EMPTYP(I)^=4 then ESR(I)=2;                                go to P20000;
P29: if WHYNOT(I)<3 then ESR(I)=3;                                         go to P20000;
P32: if DO4WKS(I)=1 then go to P20000;
if WHYNOT(I)>2 then go to P20000;
if WEEKS(I)=na then go to P33;
ESR(I)=3;                                                               go to P20000;
P33: if DO4WKS(I)>1 ! WHYNOT(I)^=na then ESR(I)=3;                      go to P20000;
P37: ESR(I)=2;                                                               go to P20000;
if ABSENT(I)^=1 then go to P43;
if WHYOFF(I)=na then go to P41;
if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P39;
if EMPTYP(I)=4 then go to P61;                                              go to P62;
P39: if WHYNOT(I)>2 then go to P61;
ESR(I)=3;                                                               go to P62;
P41: if EMPLOY(I)=0 ! EMPTYP(I)=4 then go to P61;                          go to P62;
P43: if WHYOFF(I)=na then go to P48;
if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P45;
if EMPLOY(I)=0 ! EMPTYP(I)=4 then go to P61;                                go to P62;
P45: if WEEKS(I)=na & WHYNOT(I)=na then go to P61;
if WHYNOT(I)>2 then go to P61;
ESR(I)=3;                                                               go to P62;
P48: ESR(I)=3;
if LOOK(I)^=1 then go to P53;
if DO4WKS(I)=1 then go to P61;
if WHYNOT(I)>2 then go to P61;
if DO4WKS(I)^=na ! WEEKS(I)^=na then go to P62;
if WHYNOT(I)=na then go to P61;                                              go to P62;
P53: if DO4WKS(I)=1 then go to P61;
if WHYNOT(I)>2 then go to P61;
if DO4WKS(I)=na then go to P56;
if WEEKS(I)^=na ! WHYNOT(I)^=na then go to P62;                            go to P61;
P56: if WEEKS(I)=na ! WHYNOT(I)=na then go to P61;                          go to P62;
P61: if DOING(I)<4 ! DOING(I)>6 then go to P610;
ESR(I)=DOING(I);                                                       go to P20000;
P610: ESR(I)=7;                                                       go to P20000;
P62: if DOING(I)=6 then DOING(I)=7;
P20000: go to fin;
ESR237: if WHYOFF(I)=na then go to P5019;
if WHYOFF(I)=6 ! WHYOFF(I)=7 then go to P5017;
if EMPTYP(I)^=-4 then ESR(I)=2;                                              go to fin;
P5017: if WHYNOT(I)>2 then go to fin;
ESR(I)=3;                                                               go to fin;
P5019: if EMPLOY(I)=1 & EMPTYP(I)^=4 then ESR(I)=2;                         fin;;

```

/* COLLAPSED EMPLOYMENT STATUS RECODE 1984-1993 */

```

if ESR(I)=-5 then ESRC(I)=-5;
else if ESR(I)=1 | ESR(I)=2 then ESRC(I)=1;
else if ESR(I)=3 then ESRC(I)=2;
else if ESR(I)>=4 & ESR(I)<=7 then ESRC(I)=3;
else if ESR(I)=8 then ESRC(I)=4;
else ESRC(I)=ESR(I);
end;

```

/ CHANGES BETWEEN 1993 and 1994 */*

NOTE: Users should be aware that the CPS section, on which the creation of Employment Status Recode (ESR) is based, was significantly revised in 1994-96. Hence the 1994-96 ESR variables are created using a different formula than variables in previous years (1979-1993). The formula for the 1994 ESR variable was modeled after that used to compute the MLS (Monthly Labor Status) from the actual Current Population Survey (CPS). The programs for both 1994 and 1996 are included below in their entirety. Those variables identified by reference numbers (R#####.) are those which are included on the 1979-1996 Youth CD-ROM. These are in general, questions answered by or containing valid data for, at least one respondent. The questions identified by question name (Q5.....), are those not included on the CD-ROM. These are in general, questions containing no valid data for any respondents. They remain in the formula to have it represented completely. However, the lack of valid data in those variables means that they would have virtually no effect in the formula or on the final value of ESR.

/ 1994 VARIABLES */*

```

/* The programming for the new 1994 ESR variables is done in SPSS */
/* the following lines create the var AGE */
compute AGE=37

/* the following lines construct the var HRACT1 */
do if (Q5B2_40A eq -4)
compute HRACT1=-4
else if (Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
        Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2)
compute HRACT1=-2
else if (Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995)
compute HRACT1=995
else if (Q5B2_40B eq 1 and Q5B2_41A ne -4)           compute HRACT1=Q5B2_41A
else if (Q5B2_40C eq 1 and Q5B2_41B ne -4)           compute HRACT1=Q5B2_41B
else if (Q5B2_40D eq 1 and Q5B2_41C ne -4)           compute HRACT1=Q5B2_41C
else if (Q5B2_40E eq 1 or Q5B2_40E eq 0)             compute HRACT1=Q5B2_41D
else if (Q5B_40A eq -4)                               compute HRACT1=-4
else if (Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq
        -2 or Q5B_41C eq -2 or Q5B_41D eq -2)
compute HRACT1=-2
else if (Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995)
compute HRACT1=995
else if (R(45594.) eq 1 and Q5B_41A ne -4)          compute HRACT1=Q5B_41A
else if (R(45595.) eq 1 and Q5B_41B ne -4)          compute HRACT1=Q5B_41B
else if (R(45596.) eq 1 and Q5B_41C ne -4)          compute HRACT1=Q5B_41C
else if (Q5B_40E eq 1 or Q5B_40E eq 0)               compute HRACT1=Q5B_41D
else if (R(45443.) eq -4)                            compute HRACT1=-4
else if (R(45448.) eq -1 or R(45449.) eq -1 or R(45450.) eq -1 or R(45451.) eq -1 or R(45448.) eq -2 or R(45449.) eq
        -2 or R(45450.) eq -2 or R(45451.) eq -2)
compute HRACT1=-2
else if (R(45448.) eq 995 or R(45449.) eq 995 or R(45450.) eq 995 or R(45451.) eq 995)
compute HRACT1=995
else if (R(45444.) eq 1 and R(45448.) ne -4)        compute HRACT1=R(45448.)
else if (R(45445.) eq 1 and R(45449.) ne -4)        compute HRACT1=R(45449.)
else if (R(45446.) eq 1 and R(45450.) ne -4)        compute HRACT1=R(45450.)
else if (R(45447.) eq 1 or R(45447.) eq 0)          compute HRACT1=R(45451.)
else
compute HRACT1=-3
end if

/* the following lines construct the var HRACT2 */
do if (Q5B2_42 eq -4 or Q5B2_42 eq 0)                compute HRACT2=-4
else if (Q5B2_43A eq -1 or Q5B2_43B eq -1OR Q5B2_43A eq -2 or Q5B2_43B eq -2)      compute HRACT2=-2

```

```

else if (Q5B2_43A eq 995 or Q5B2_43B eq 995)           compute HRACT2=995
else if (Q5B2_43 eq 1)                                 compute HRACT2=Q5B2_43A
else if (Q5B2_42 eq 1 and Q5B2_43 eq 0)               compute HRACT2=Q5B2_43B
else if (Q5B_42 eq -4 or Q5B_42 eq 0)                 compute HRACT2=-4
else if (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2)   compute HRACT2=-2
else if (Q5B_43A eq 995 or Q5B_43B eq 995)           compute HRACT2=995
else if (Q5B_43 eq 1)                                 compute HRACT2=Q5B_43A
else if (Q5B_42 eq 1 and Q5B_43 eq 0)               compute HRACT2=Q5B_43B
else if (R(45452.) eq -4 or R(45452.) eq 0)          compute HRACT2=-4
else if (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2)   compute HRACT2=-2
else if (R(45454.) eq 995 or R(45455.) eq 995)       compute HRACT2=995
else if (R(45453.) eq 1)                             compute HRACT2=R(45454.)
else if (R(45452.) eq 1 and R(45453.) eq 0)          compute HRACT2=R(45455.)
else                                                 compute HRACT2=-3
end if

/* the following lines create the var HRACTT */
do if (Q5B2_40A eq -4)
compute HRACTT=-4
else if ((Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
         Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2) and (Q5B2_43A eq -1 or Q5B2_43B eq -1 or
         Q5B2_43A eq -2 or Q5B2_43B eq -2))
compute HRACTT=-2
else if (((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and (Q5B2_43A eq
         995 or Q5B2_43B eq 995)))
compute HRACTT=995
else if (((Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
         Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2) and (Q5B2_43A eq 995 or Q5B2_43B eq 995)))
compute HRACTT=995
else if (((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995 or Q5B2_41A eq
         995
         or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and (Q5B2_43A eq -1 or Q5B2_43B eq -1
         or Q5B2_43A eq -2 or Q5B2_43B eq -2))
compute HRACTT=995
else if (((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43A eq -1
         or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41A
else if (((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43A eq -1
         or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41B
else if (((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43A eq -1
         or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41C
else if (((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43A eq
         -1 or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41D
else if (((Q5B2_43A ne -4 and Q5B2_43A ne -1 and Q5B2_43A ne -2 and Q5B2_43A ne 995) and (Q5B2_41A eq -1
         or Q5B2_41A eq -2 or Q5B2_41A eq 995 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41B eq 995 or
         Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41C eq 995 or Q5B2_41D eq -1 or Q5B2_41D eq -2 or
         Q5B2_41D eq 995))
compute HRACTT=Q5B2_43A
else if (((Q5B2_43B ne -4 and Q5B2_43B ne -1 and Q5B2_43B ne -2 and Q5B2_43B ne 995) and (Q5B2_41A eq -1
         or Q5B2_41A eq -2 or Q5B2_41A eq 995 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41B eq 995 or
         Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41C eq 995 or Q5B2_41D eq -1 or Q5B2_41D eq -2 or
         Q5B2_41D eq 995))
compute HRACTT=Q5B2_43B
else if (((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43A ne -1
         and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
compute HRACTT=Q5B2_41A+Q5B2_43A
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43B ne -1
        and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41A+Q5B2_43B
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43A ne -1
        and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41B+Q5B2_43A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43B ne -1
        and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41B+Q5B2_43B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43A ne -1
        and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41C+Q5B2_43A
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43B ne -1
        and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41C+Q5B2_43B
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43A ne
        -1 and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41D+Q5B2_43A
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43B ne
        -1 and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41D+Q5B2_43B
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and Q5B2_43A eq -4
        and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and Q5B2_43A eq -4
        and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and Q5B2_43A eq -4
        and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41C
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and Q5B2_43A eq -4
        and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41D
else if ((Q5B2_41A eq -1 or Q5B2_41A eq -2 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41C eq -1 or
        Q5B2_41C eq -2 or Q5B2_41D eq -1 or Q5B2_41D eq -2) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=-2
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and Q5B2_43A eq -4
        and Q5B2_43B eq -4)
compute HRACTT=995
else if (Q5B_40A eq -4)
compute HRACTT=-4
else if ((Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq
        -2 or Q5B_41C eq -2 or Q5B_41D eq -2) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or
        Q5B_43B eq -2))
compute HRACTT=-2
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq 995 or
        Q5B_43B eq 995))
compute HRACTT=995
else if ((Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq
        -2 or Q5B_41C eq -2 or Q5B_41D eq -2) and (Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=995
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995 or Q5B_41A eq 995 or
        Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or
        Q5B_43A eq -2 or Q5B_43B eq -2))
compute HRACTT=995
```

```

else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A eq -1 or
        Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A eq -1 or
        Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A eq -1 or
        Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41C
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A eq -1 or
        Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41D
else if ((Q5B_43A ne -4 and Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995) and (Q5B_41A eq -1 or
        Q5B_41A eq -2 or Q5B_41A eq 995 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41B eq 995 or Q5B_41C
        eq -1 or Q5B_41C eq -2 or Q5B_41C eq 995 or Q5B_41D eq -1 or Q5B_41D eq -2 or Q5B_41D eq 995))
compute HRACTT=Q5B_43A
else if ((Q5B_43B ne -4 and Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995) and (Q5B_41A eq -1 or
        Q5B_41A eq -2 or Q5B_41A eq 995 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41B eq 995 or Q5B_41C
        eq -1 or Q5B_41C eq -2 or Q5B_41C eq 995 or Q5B_41D eq -1 or Q5B_41D eq -2 or Q5B_41D eq 995))
compute HRACTT=Q5B_43B
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41A+Q5B_43A
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41A+Q5B_43B
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41B+Q5B_43A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41B+Q5B_43B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41C+Q5B_43A
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41C+Q5B_43B
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41D+Q5B_43A
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41D+Q5B_43B
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41C
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41D

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
else if ((Q5B_41A eq -1 or Q5B_41A eq -2 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41C eq -1 or Q5B_41C eq -2 or Q5B_41D eq -1 or Q5B_41D eq -2) and Q5B_43A eq -4 and Q5B_43B eq -4)
compute HRACTT=-2
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and Q5B_43A eq -4 and Q5B_43B eq -4)
compute HRACTT=995
else if (R(45443.) eq -4)
compute HRACTT=-4
else if ((R(45448.) eq -1 or R(45449.) eq -1 or R(45450.) eq -1 or R(45451.) eq -1 or R(45448.) eq -2 or R(45449.) eq -2 or R(45450.) eq -2 or R(45451.) eq -2) and (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2))
compute HRACTT=-2
else if ((R(45448.) eq 995 or R(45449.) eq 995 or R(45450.) eq 995 or R(45451.) eq 995) and (R(45454.) eq 995 or R(45455.) eq 995))
compute HRACTT=995
else if ((R(45448.) eq -1 or R(45449.) eq -1 or R(45450.) eq -1 or R(45451.) eq -1 or R(45448.) eq -2 or R(45449.) eq -2 or R(45450.) eq -2 or R(45451.) eq -2) and (R(45454.) eq 995 or R(45455.) eq 995))
compute HRACTT=995
else if ((R(45448.) eq 995 or R(45449.) eq 995 or R(45450.) eq 995 or R(45451.) eq 995 or R(45448.) eq 995 or R(45449.) eq 995 or R(45450.) eq 995 or R(45451.) eq 995) and (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2))
compute HRACTT=995
else if ((R(45448.) ne -4 and R(45448.) ne -1 and R(45448.) ne -2 and R(45448.) ne 995) and (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2 or R(45454.) eq 995 or R(45455.) eq 995))
compute HRACTT=R(45448.)
else if ((R(45449.) ne -4 and R(45449.) ne -1 and R(45449.) ne -2 and R(45449.) ne 995) and (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2 or R(45454.) eq 995 or R(45455.) eq 995))
compute HRACTT=R(45449.)
else if ((R(45450.) ne -4 and R(45450.) ne -1 and R(45450.) ne -2 and R(45450.) ne 995) and (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2 or R(45454.) eq 995 or R(45455.) eq 995))
compute HRACTT=R(45450.)
else if ((R(45451.) ne -4 and R(45451.) ne -1 and R(45451.) ne -2 and R(45451.) ne 995) and (R(45454.) eq -1 or R(45455.) eq -1 or R(45454.) eq -2 or R(45455.) eq -2 or R(45454.) eq 995 or R(45455.) eq 995))
compute HRACTT=R(45451.)
else if ((R(45454.) ne -4 and R(45454.) ne -1 and R(45454.) ne -2 and R(45454.) ne 995) and (R(45448.) eq -1 or R(45448.) eq -2 or R(45448.) eq 995 or R(45449.) eq -1 or R(45449.) eq -2 or R(45449.) eq 995 or R(45450.) eq -1 or R(45450.) eq -2 or R(45450.) eq 995 or R(45451.) eq -1 or R(45451.) eq -2 or R(45451.) eq 995))
compute HRACTT=R(45454.)
else if ((R(45455.) ne -4 and R(45455.) ne -1 and R(45455.) ne -2 and R(45455.) ne 995) and R(45448.) eq -1 or R(45448.) eq -2 or R(45448.) eq 995 or R(45449.) eq -1 or R(45449.) eq -2 or R(45449.) eq 995 or R(45450.) eq -1 or R(45450.) eq -2 or R(45450.) eq 995 or R(45451.) eq -1 or R(45451.) eq -2 or R(45451.) eq 995))
compute HRACTT=R(45455.)
else if ((R(45448.) ne -4 and R(45448.) ne -1 and R(45448.) ne -2 and R(45448.) ne 995) and (R(45454.) ne -1 and R(45454.) ne -2 and R(45454.) ne 995 and R(45454.) ne -4))
compute HRACTT=R(45448.)+R(45454.)
else if ((R(45448.) ne -4 and R(45448.) ne -1 and R(45448.) ne -2 and R(45448.) ne 995) and (R(45455.) ne -1 and R(45455.) ne -2 and R(45455.) ne 995 and R(45455.) ne -4))
compute HRACTT=R(45448.)+R(45455.)
else if ((R(45449.) ne -4 and R(45449.) ne -1 and R(45449.) ne -2 and R(45449.) ne 995) and (R(45454.) ne -1 and R(45454.) ne -2 and R(45454.) ne 995 and R(45454.) ne -4))
compute HRACTT=R(45449.)+R(45454.)
else if ((R(45449.) ne -4 and R(45449.) ne -1 and R(45449.) ne -2 and R(45449.) ne 995) and (R(45455.) ne -1 and R(45455.) ne -2 and R(45455.) ne 995 and R(45455.) ne -4))
compute HRACTT=R(45449.)+R(45455.)
else if ((R(45450.) ne -4 and R(45450.) ne -1 and R(45450.) ne -2 and R(45450.) ne 995) and (R(45454.) ne -1 and R(45454.) ne -2 and R(45454.) ne 995 and R(45454.) ne -4))
compute HRACTT=R(45450.)+R(45454.)
```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
else if ((R(45450.) ne -4 and R(45450.) ne -1 and R(45450.) ne -2 and R(45450.) ne 995) and (R(45455.) ne -1 and  
    R(45455.) ne -2 and R(45455.) ne 995 and R(45455.) ne -4))  
compute HRACTT=R(45450.)+R(45455.)  
else if ((R(45451.) ne -4 and R(45451.) ne -1 and R(45451.) ne -2 and R(45451.) ne 995) and (R(45454.) ne -1 and  
    R(45454.) ne -2 and R(45454.) ne 995 and R(45454.) ne -4))  
compute HRACTT=R(45451.)+R(45454.)  
else if ((R(45451.) ne -4 and R(45451.) ne -1 and R(45451.) ne -2 and R(45451.) ne 995) and (R(45455.) ne -1 and  
    R(45455.) ne -2 and R(45455.) ne 995 and R(45455.) ne -4))  
compute HRACTT=R(45451.)+R(45455.)  
else if ((R(45448.) ne -4 and R(45448.) ne -1 and R(45448.) ne -2 and R(45448.) ne 995) and R(45454.) eq -4 and  
    R(45455.) eq -4)  
compute HRACTT=R(45448.)  
else if ((R(45449.) ne -4 and R(45449.) ne -1 and R(45449.) ne -2 and R(45449.) ne 995) and R(45454.) eq -4 and  
    R(45455.) eq -4)  
compute HRACTT=R(45449.)  
else if ((R(45450.) ne -4 and R(45450.) ne -1 and R(45450.) ne -2 and R(45450.) ne 995) and R(45454.) eq -4 and  
    R(45455.) eq -4)  
compute HRACTT=R(45450.)  
else if ((R(45451.) ne -4 and R(45451.) ne -1 and R(45451.) ne -2 and R(45451.) ne 995) and R(45454.) eq -4 and  
    R(45455.) eq -4)  
compute HRACTT=R(45451.)  
else if ((R(45448.) eq -1 or R(45448.) eq -2 or R(45449.) eq -1 or R(45449.) eq -2 or R(45450.) eq -1 or R(45450.) eq  
    -2 or R(45451.) eq -1 or R(45451.) eq -2) and R(45454.) eq -4 and R(45455.) eq -4)  
compute HRACTT=-2  
else if ((R(45448.) eq 995 or R(45449.) eq 995 or R(45450.) eq 995 or R(45451.) eq 995) and R(45454.) eq -4 and  
    R(45455.) eq -4)  
compute HRACTT=995  
else  
compute HRACTT=-3  
end if  
  
/* the following lines creates the var BUS2 */  
do if (R(45368.) eq -4) compute BUS2=-4  
else if (R(45368.) eq -1) compute BUS2=-1  
else if (R(45368.) eq -2) compute BUS2=-2  
else if (R(45369.) eq -1 or R(45370.) eq -1) compute BUS2=-1  
else if (R(45369.) eq -2 or R(45370.) eq -2) compute BUS2=-2  
else if (R(45369.) eq 1 or R(45370.) eq 1) compute BUS2=1  
else if (R(45370.) eq 0) compute BUS2=2 /* this should normally be 0 */  
else compute BUS2=-3  
end if  
  
/* the following lines create the var HRUSL1 */  
do if (R(45665.) eq -4) compute HRUSL1=-4  
else if (R(45666.) eq -1 or R(45667.) eq -1 or R(45666.) eq -2 or R(45667.) eq -2) compute HRUSL1=-2  
else if (R(45666.) eq 995 or R(45667.) eq 995) compute HRUSL1=995  
else if (R(45666.) eq 1) compute HRUSL1=Q5B2_26B  
else if (R(45667.) eq 1) compute HRUSL1=Q5B2_26D  
else if (R(45582.) eq -4) compute HRUSL1=-4  
else if (Q5B_26A eq -1 or R(45583.) eq -1 or Q5B_26A eq -2 or R(45583.) eq -2) compute HRUSL1=-2  
else if (Q5B_26A eq 995 or R(45583.) eq 995) compute HRUSL1=995  
else if (Q5B_26A eq 1) compute HRUSL1=Q5B_26B  
else if (R(45583.) eq 1) compute HRUSL1=R(45584.)  
else if (R(45399.) eq -4) compute HRUSL1=-4  
else if (R(45400.) eq -1 or R(45402.) eq -1 or R(45400.) eq -2 or R(45402.) eq -2) compute HRUSL1=-2  
else if (R(45400.) eq 995 or R(45402.) eq 995) compute HRUSL1=995  
else if (R(45400.) eq 1) compute HRUSL1=R(45401.)
```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

else if (R(45402.) eq 1) compute HRUSL1=R(45403.)
else compute HRUSL1=-3
end if

/* the following lines create the var HRUSL2 */
do if (Q5B2_28A eq -4) compute HRUSL2=-4
else if (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B eq -2 or Q5B2_28D eq -2) compute HRUSL2=-2
else if (Q5B2_28B eq 995 or Q5B2_28D eq 995) compute HRUSL2=995
else if (Q5B2_28B eq 1) compute HRUSL2=Q5B2_28C
else if (Q5B2_28D eq 1) compute HRUSL2=Q5B2_28E
else if (Q5B2_28A eq -4) compute HRUSL2=-4
else if (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2) compute HRUSL2=-2
else if (Q5B_28B eq 995 or Q5B_28D eq 995) compute HRUSL2=995
else if (Q5B_28B eq 1) compute HRUSL2=Q5B_28C
else if (Q5B_28D eq 1) compute HRUSL2=Q5B_28E
else if (R(45405.) eq -4) compute HRUSL2=-4
else if (R(45406.) eq -1 or R(45408.) eq -1 or R(45406.) eq -2 or R(45408.) eq -2) compute HRUSL2=-2
else if (R(45406.) eq 995 or R(45408.) eq 995) compute HRUSL2=995
else if (R(45406.) eq 1) compute HRUSL2=Q5_28C
else if (R(45408.) eq 1) compute HRUSL2=Q5_28E
else compute HRUSL2=-3
end if

/* the following lines create the var HRUSLT */
do if (R(45665.) eq -4) compute HRUSLT=-4
compute HRUSLT=-4
else if ((R(45666.) eq -1 or R(45667.) eq -1 or R(45666.) eq -2 or R(45667.) eq -2) and (Q5B2_28B eq -1 or
      Q5B2_28D eq -1 or Q5B2_28B eq -2 or Q5B2_28D eq -2)) compute HRUSLT=-2
else if ((R(45666.) eq 995 or R(45667.) eq 995) and (Q5B2_28B eq 995 or Q5B2_28D eq 995)) compute HRUSLT=995
else if ((R(45666.) eq -1 or R(45667.) eq -1 or R(45666.) eq -2 or R(45667.) eq -2) and (Q5B2_28B eq 995 or
      Q5B2_28D eq 995)) compute HRUSLT=995
else if ((R(45666.) eq 995 or R(45667.) eq 995) and (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B eq -2 or
      Q5B2_28D eq -2)) compute HRUSLT=995
else if ((R(45666.) eq 995 or R(45667.) eq 995) and Q5B2_27 ne 1) compute HRUSLT=995
else if ((R(45666.) eq -1 or R(45667.) eq -2 or R(45667.) eq -2) and Q5B2_27 ne 1) compute HRUSLT=-2
else if (R(45666.) eq 1 and (Q5B2_28B eq -1 or Q5B2_28B eq -2 or Q5B2_28B eq 995 or Q5B2_28D eq -1 or
      Q5B2_28D eq -2 or Q5B2_28D eq 995)) compute HRUSLT=Q5B2_26B
else if (R(45667.) eq 1 and (Q5B2_28B eq -1 or Q5B2_28B eq -2 or Q5B2_28B eq 995 or Q5B2_28D eq -1 or
      Q5B2_28D eq -2 or Q5B2_28D eq 995)) compute HRUSLT=Q5B2_26D
else if (Q5B2_28B eq 1 and (R(45666.) eq -1 or R(45666.) eq -2 or R(45666.) eq 995 or R(45667.) eq -1 or R(45667.) eq -2 or R(45667.) eq 995)) compute HRUSLT=Q5B2_28C
else if (Q5B2_28D eq 1 and (R(45666.) eq -1 or R(45666.) eq -2 or R(45666.) eq 995 or R(45667.) eq -1 or
      R(45667.) eq -2 or R(45667.) eq 995)) compute HRUSLT=Q5B2_28E
else if (R(45666.) eq 1 and Q5B2_27 ne 1) compute HRUSLT=Q5B2_26B
else if (R(45667.) eq 1 and Q5B2_27 ne 1) compute HRUSLT=Q5B2_26D
else if (R(45666.) eq 1 and Q5B2_28B eq 1) compute HRUSLT=Q5B2_26B+Q5B2_28C
else if (R(45666.) eq 1 and Q5B2_28D eq 1) compute HRUSLT=Q5B2_26B+Q5B2_28E

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

else if (R(45667.) eq 1 and Q5B2_28B eq 1) compute HRUSLT=Q5B2_26D+Q5B2_28C
else if (R(45667.) eq 1 and Q5B2_28D eq 1) compute HRUSLT=R(45667.)+Q5B2_28E
else if (R(45582.) eq -4) compute HRUSLT=-4
else if ((Q5B_26A eq -1 or R(45583.) eq -1 or Q5B_26A eq -2 or R(45583.) eq -2) and (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2))
compute HRUSLT=-2
else if ((Q5B_26A eq 995 or R(45583.) eq 995) and (Q5B_28B eq 995 or Q5B_28D eq 995))
compute HRUSLT=995
else if ((Q5B_26A eq -1 or R(45583.) eq -1 or Q5B_26A eq -2 or R(45583.) eq -2) and (Q5B_28B eq 995 or Q5B_28D eq 995))
compute HRUSLT=995
else if ((Q5B_26A eq 995 or R(45583.) eq 995) and (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2))
compute HRUSLT=995
else if ((Q5B_26A eq 995 or R(45583.) eq 995) and R(45585.) ne 1)
compute HRUSLT=995
else if ((Q5B_26A eq -1 or R(45583.) eq -1 or Q5B_26A eq -2 or R(45583.) eq -2) and R(45585.) ne 1)
compute HRUSLT=-2
else if (Q5B_26A eq 1 and (Q5B_28B eq -1 or Q5B_28B eq -2 or Q5B_28B eq 995 or Q5B_28D eq -1 or Q5B_28D eq -2 or Q5B_28D eq 995))
compute HRUSLT=Q5B_26B
else if (R(45583.) eq 1 and (Q5B_28B eq -1 or Q5B_28B eq -2 or Q5B_28B eq 995 or Q5B_28D eq -1 or Q5B_28D eq -2 or Q5B_28D eq 995))
compute HRUSLT=R(45584.)
else if (Q5B_28B eq 1 and (Q5B_26A eq -1 or Q5B_26A eq -2 or Q5B_26A eq 995 or R(45583.) eq -1 or R(45583.) eq -2 or R(45583.) eq 995))
compute HRUSLT=Q5B_28C
else if (Q5B_28D eq 1 and (Q5B_26A eq -1 or Q5B_26A eq -2 or Q5B_26A eq 995 or R(45583.) eq -1 or R(45583.) eq -2 or R(45583.) eq 995))
compute HRUSLT=Q5B_28D
else if (Q5B_26A eq 1 and R(45585.) ne 1) compute HRUSLT=Q5B_26B
else if (R(45583.) eq 1 and R(45585.) ne 1) compute HRUSLT=R(45584.)
else if (Q5B_26A eq 1 and Q5B_28B eq 1) compute HRUSLT=Q5B_26B+Q5B_28C
else if (Q5B_26A eq 1 and Q5B_28D eq 1) compute HRUSLT=Q5B_26B+Q5B_28E
else if (R(45583.) eq 1 and Q5B_28B eq 1) compute HRUSLT=R(45584.)+Q5B_28C
else if (R(45583.) eq 1 and Q5B_28D eq 1) compute HRUSLT=R(45583.)+Q5B_28E
else if (R(45399.) eq -4) compute HRUSLT=-4
else if ((R(45400.) eq -1 or R(45402.) eq -1 or R(45400.) eq -2 or R(45402.) eq -2) and (R(45406.) eq -1 or R(45408.) eq -1 or R(45406.) eq -2 or R(45408.) eq -2))
compute HRUSLT=-2
else if ((R(45400.) eq 995 or R(45402.) eq 995) and (R(45406.) eq 995 or R(45408.) eq 995))
compute HRUSLT=995
else if ((R(45400.) eq -1 or R(45402.) eq -1 or R(45400.) eq -2 or R(45402.) eq -2) and (R(45406.) eq 995 or R(45408.) eq 995))
compute HRUSLT=995
else if ((R(45400.) eq 995 or R(45402.) eq 995) and (R(45406.) eq -1 or R(45408.) eq -1 or R(45406.) eq -2 or R(45408.) eq -2))
compute HRUSLT=995
else if ((R(45400.) eq 995 or R(45402.) eq 995) and R(45404.) ne 1)
compute HRUSLT=995
else if ((R(45400.) eq -1 or R(45402.) eq -1 or R(45400.) eq -2 or R(45402.) eq -2) and R(45404.) ne 1)
compute HRUSLT=-2
else if (R(45400.) eq 1 and (R(45406.) eq -1 or R(45406.) eq -2 or R(45406.) eq 995 or R(45408.) eq -1 or R(45408.) eq -2 or R(45408.) eq 995))
compute HRUSLT=R(45401.)
else if (R(45402.) eq 1 and (R(45406.) eq -1 or R(45406.) eq -2 or R(45406.) eq 995 or R(45408.) eq -1 or R(45408.) eq -2 or R(45408.) eq 995))
compute HRUSLT=R(45402.)

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
compute HRUSLT=R(45403.)
else if (R(45406.) eq 1 and (R(45400.) eq -1 or R(45400.) eq -2 or R(45400.) eq 995 or R(45402.) eq -1 or R(45402.) eq -2 or R(45402.) eq 995))
compute HRUSLT=R(45407.)
else if (R(45408.) eq 1 and (R(45400.) eq -1 or R(45400.) eq -2 or R(45400.) eq 995 or R(45402.) eq -1 or R(45402.) eq -2 or R(45402.) eq 995))
compute HRUSLT=R(45409.)
else if (R(45400.) eq 1 and R(45404.) ne 1) compute HRUSLT=R(45401.)
else if (R(45402.) eq 1 and R(45404.) ne 1) compute HRUSLT=R(45403.)
else if (R(45400.) eq 1 and R(45406.) eq 1) compute HRUSLT=R(45401.)+R(45407.)
else if (R(45400.) eq 1 and R(45408.) eq 1) compute HRUSLT=R(45401.)+R(45409.)
else if (R(45402.) eq 1 and R(45406.) eq 1) compute HRUSLT=R(45403.)+R(45407.)
else if (R(45402.) eq 1 and R(45408.) eq 1) compute HRUSLT=R(45402.)+R(45409.)
else
compute HRUSLT=-3
end if

/* the following lines create the var HRFTPPT */
do if (Q5B2_30C eq -4)           compute HRFTPPT=-4
else if (Q5B2_30F eq -1 or Q5B2_30G eq -1 or Q5B2_30H eq -1 or Q5B2_30I eq -1)   compute HRFTPPT=-1
else if (Q5B2_30F eq -2 or Q5B2_30G eq -2 or Q5B2_30H eq -2 or Q5B2_30I eq -2)   compute HRFTPPT=-2
else if (Q5B2_30F ne -4)          compute HRFTPPT=Q5B2_30F
else if (Q5B2_30G ne -4)          compute HRFTPPT=Q5B2_30G
else if (Q5B2_30H ne -4)          compute HRFTPPT=Q5B2_30H
else if (Q5B2_30I ne -4)          compute HRFTPPT=Q5B2_30I
else if (Q5B_30C eq -4)           compute HRFTPPT=-4
else if (Q5B_30F eq -1 or Q5B_30G eq -1 or Q5B_30H eq -1 or Q5B_30I eq -1)      compute HRFTPPT=-1
else if (Q5B_30F eq -2 or Q5B_30G eq -2 or Q5B_30H eq -2 or Q5B_30I eq -2)      compute HRFTPPT=-2
else if (Q5B_30F ne -4)          compute HRFTPPT=Q5B_30F
else if (Q5B_30G ne -4)          compute HRFTPPT=Q5B_30G
else if (Q5B_30H ne -4)          compute HRFTPPT=Q5B_30H
else if (Q5B_30I ne -4)          compute HRFTPPT=Q5B_30I
else if (R(45420.) eq -4)         compute HRFTPPT=-4
else if (Q5_30F eq -1 or R(45423.) eq -1 or R(45424.) eq -1 or R(45425.) eq -1)  compute HRFTPPT=-1
else if (Q5_30F eq -2 or R(45423.) eq -2 or R(45424.) eq -2 or R(45425.) eq -2)  compute HRFTPPT=-2
else if (Q5_30F ne -4)          compute HRFTPPT=Q5_30F
else if (R(45423.) ne -4)        compute HRFTPPT=R(45423.)
else if (R(45424.) ne -4)        compute HRFTPPT=R(45424.)
else if (R(45425.) ne -4)        compute HRFTPPT=R(45425.)
else
compute HRFTPPT=-3
end if

/* the following lines create the var HRWANT */
do if (Q5B2_33 ne -4)           compute HRWANT=Q5B2_33
else if (Q5B_33 ne -4)          compute HRWANT=Q5B_33
else
compute HRWANT=R(45436.)
end if

/* the following lines create the var HRRSN1 */
do if (Q5B2_34 ne -4)           compute HRRSN1=Q5B2_34
else if (Q5B_34 ne -4)          compute HRRSN1=Q5B_34
else
compute HRRSN1=R(45437.)
end if

/* the following lines create the var HRCK7 */
compute HRCK7=-4
do if ((BUS2 eq 2 or BUS2 eq -2 or BUS2 eq -1) and (HRACT1 LT 15 or HRACT1 eq -2))
compute HRCK7=1
```

```

else if ((BUS2 eq 2 or BUS2 eq -2 or BUS2 eq -1) and HRACT1 GE 15)
compute HRCK7=2
else if ((HRUSLT GE 35 or HRFTPT eq 1) and HRACTT LT 35 and (HRACT1 ne -1 or HRACT1 ne -2 or
          HRACT2 ne -1 or HRACT2 ne -2))
compute HRCK7=3
else if (HRWANT eq 1 and HRACTT LT 35 and (HRRSN1 eq 1 or HRRSN1 eq 2 or HRRSN1 eq 3))
compute HRCK7=4
else
compute HRCK7=5
end if

/* the following lines create the var ABSOT */
do if (HRACT1 eq 0 and HRACT2 eq 0)  compute ABSOT=1
else                                compute ABSOT=0      end if

/* the following lines create the var HRCK6 */
compute HRCK6=-4
do if ((HRACT1 eq 0 and HRACT2 eq 0) and (BUS2 eq 2 or BUS2 eq -1 or BUS2 eq -2))
compute HRCK6=1
else if (ABSOT eq 1)                  compute HRCK6=2
else                                compute HRCK6=3      end if

/* the following lines create the var WK */
compute WK=-4
do if (R(45364.) ne -4)              compute WK=R(45364.)
else if (R(45365.) ne -4)           compute WK=R(45365.)    end if

/* the following lines create the var RETOT */
do if (WK eq 3)                      compute RETOT=1
else                                compute RETOT=0      end if

/* the following lines rename the vars */
compute BUS1=R(45367.)               compute RET1=R(45374.)
compute IO1COW=R(45360.)             compute NLFACT=R(45737.)

/* the following lines create additional variables */
do if (Q5B2_22 ne -4)                compute ABSPD=Q5B2_22
else if (R(45577.) ne -4)            compute ABSPD=R(45577.)
else                                compute ABSPD=R(45393.)    end if

do if (Q5B2_65 ne -4)                compute LKAVL=Q5B2_65
else if (R(45647.) ne -4)            compute LKAVL=R(45647.)
else                                compute LKAVL=R(45539.)    end if

do if (Q5B2_51 ne -4)                compute LAYAVL=Q5B2_51
else if (Q5B_51 ne -4)               compute LAYAVL=Q5B_51
else                                compute LAYAVL=R(45470.)    end if

do if (Q5B2_66 ne -4)                compute LKAVR=Q5B2_66
else if (Q5B_66 ne -4)               compute LKAVR=Q5B_66
else                                compute LKAVR=R(45540.)    end if

do if (Q5B2_52 ne -4)                compute LAYAVR=Q5B2_52
else if (Q5B_52 ne -4)               compute LAYAVR=Q5B_52
else                                compute LAYAVR=R(45471.)    end if

do if (Q5B2_60 ne -4)                compute LK=Q5B2_60

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
else if (R(45600.) ne -4)      compute LK=R(45600.)
else                           compute LK=R(45480.)           end if

do if (Q5B2_11 ne -4)          compute DIS1=Q5B2_11
else if (R(45562.) ne -4)     compute DIS1=R(45562.)
else                           compute DIS1=R(45375.)         end if

do if (Q5B2_12 ne -4)          compute DIS2=Q5B2_12
else if (R(45563.) ne -4)     compute DIS2=R(45563.)
else                           compute DIS2=R(45376.)         end if

/* the following lines create the var DWWNTO */
do if (RET1 eq 1)              compute DWWNTO=1

else if (Q5B2_78 eq 1)          compute DWWNTO=1
else if (Q5B2_78 eq 0)          compute DWWNTO=2
else if (Q5B2_78 eq -2)        compute DWWNTO=-2
else if (Q5B2_78 eq -1)        compute DWWNTO=-1
else if (Q5B2_78 eq 3)          compute DWWNTO=3
else if (Q5B2_78 eq 4)          compute DWWNTO=4
else if (R(45662.) eq 1)        compute DWWNTO=1
else if (R(45662.) eq 0)        compute DWWNTO=2
else if (R(45662.) eq -2)      compute DWWNTO=-2
else if (R(45662.) eq -1)      compute DWWNTO=-1
else if (R(45662.) eq 3)        compute DWWNTO=3
else if (R(45662.) eq 4)        compute DWWNTO=4
else if (R(45558.) eq 1)        compute DWWNTO=1
else if (R(45558.) eq 0)        compute DWWNTO=2
else if (R(45558.) eq -2)      compute DWWNTO=-2
else if (R(45558.) eq -1)      compute DWWNTO=-1
else if (R(45558.) eq 3)        compute DWWNTO=3
else if (R(45558.) eq 4)        compute DWWNTO=4
else                           compute DWWNTO=-3           end if

/* the following lines create the var DIS */
do if (Q5_9A4 eq -4)           compute DIS=-4
else if (Q5_9D eq -1 or Q5_9E eq -1 or Q5_9F eq -1 or Q5_9G eq -1) or (Q5_9D eq -2 or Q5_9E eq -2 or Q5_9F eq -2 or Q5_9G eq -2)
compute DIS=-2
else if (Q5_9A4 eq 1 and Q5_9D eq 1)    compute DIS=1
else if (Q5_9D eq 0)             compute DIS=0
else if (Q5_9A5 eq 1 and Q5_9E eq 1)    compute DIS=1
else if (Q5_9E eq 0)             compute DIS=0
else if (Q5_9A6 eq 1 and Q5_9F eq 1)    compute DIS=1
else if (Q5_9F eq 0)             compute DIS=0
else if (Q5_9A7 eq 1 and Q5_9G eq 1)    compute DIS=1
else if (Q5_9G eq 0)             compute DIS=0
else                           compute DIS=-3           end if

/* the following lines create IO1COW */
do if (R(45360.) eq 1)           compute IO1COW=8
else                           compute IO1COW=0           end if

/* the following lines create the ESR var approximating MLR */
do if ((WK eq 1 and (HRCK6 eq 3 or HRCK6 eq -4 or HRCK6 eq 99 or HRCK6 eq 999)) or (BUS1 eq 1 and (HRCK7 eq 2 or HRCK7 eq 3 or HRCK7 eq 4 or HRCK7 eq 5 or HRCK7 eq -4)))
```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
compute ESR=1
else if (ABSPD eq 1 or ABSPD eq 0 or ABSPD eq -1 or ABSPD eq -2)
compute ESR=2
else if ((LAYAVL eq 1 or LAYAVL eq -1 or LAYAVL eq -2) or LAYAVR eq 1)
compute ESR=3
else if ((LKAVL eq 1 or LKAVL eq -1 or LKAVL eq -2) or (LKAVR eq 1 or LKAVR eq 2))
compute ESR=4
else if ((WK eq 3 and AGE GE 50 and RET1 eq 1 and LK eq 1 and LKAVL eq 0 and (LKAVR eq 3 or LKAVR eq 4 or
LKAVR eq -1 or LKAVR eq -2)) or (RET1 eq 0 or LK eq 3 or DWWNTO eq 3) or NLFACI eq 5 or
RETOT eq 1)
compute ESR=5
else if (DIS1 eq 1 or DIS2 eq 1 or DIS eq 1)          compute ESR=6
else if (R(45360.) eq 1)                            compute ESR=8
else                                              compute ESR=7
end if

ESR(6)=R(15199.);      ESR(7)=R(18906.);      ESR(8)=R(22577.);
ESR(9)=R(24451.);      ESR(10)=R(28706.);     ESR(11)=R(30743.);
ESR(12)=R(34010.);     ESR(13)=R(36564.);     ESR(14)=R(40069.);
ESR(15)=R(44180.);     ESR(16)=R(50810.);     ESRC(6)=R(15199.01);
ESRC(7)=R(18906.01);   ESRC(8)=R(22577.01);   ESRC(9)=R(24451.01);
ESRC(10)=R(28707.);    ESRC(11)=R(30744.);    ESRC(12)=R(34011.);
EARC(13)=R(36565.);    EARC(14)=R(40070.);    ESRC(15)=R(44181.);
ESRC(16)=R(50811.);
```

```

/* 1996 VARIABLES */

/* The programming for the new 1996 ESR variables is done in SPSS */
/* the following lines create the var AGE */
compute AGE=37

/* the following lines construct the var HRACT1 */
do if (Q5B2_40A eq -4)
compute HRACT1=-4
else if (Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
        Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2)
compute HRACT1=-2
else if (Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995)
compute HRACT1=995
else if (Q5B2_40B eq 1 and Q5B2_41A ne -4)           compute HRACT1=Q5B2_41A
else if (Q5B2_40C eq 1 and Q5B2_41B ne -4)           compute HRACT1=Q5B2_41B
else if (Q5B2_40D eq 1 and Q5B2_41C ne -4)           compute HRACT1=Q5B2_41C
else if (Q5B2_40E eq 1 or Q5B2_40E eq 0)             compute HRACT1=Q5B2_41D
else if (Q5B2_40A eq -4)                             compute HRACT1=-4
else if (Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq
        -2 or Q5B_41C eq -2 or Q5B_41D eq -2)
compute HRACT1=-2
else if (Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995)
compute HRACT1=995
else if (Q5BCK_40B eq 1 and Q5B_41A ne -4)          compute HRACT1=Q5B_41A
else if (Q5BCK_40C eq 1 and Q5B_41B ne -4)          compute HRACT1=Q5B_41B
else if (Q5BCK_40D eq 1 and Q5B_41C ne -4)          compute HRACT1=Q5B_41C
else if (Q5B_40E eq 1 or Q5B_40E eq 0)               compute HRACT1=Q5B_41D
else if (R(52404.) eq -4)                            compute HRACT1=-4
else if (Q5_41A eq -1 or Q5_41B eq -1 or Q5_41C eq -1 or Q5_41D eq -1 or Q5_41A eq -2 or Q5_41B eq -2 or
        Q5_41C eq -2 or Q5_41D eq -2)
compute HRACT1=-2
else if (Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995)
compute HRACT1=995
else if (Q5_40B eq 1 and Q5_41A ne -4)              compute HRACT1= Q5_41A
else if (Q5_40C eq 1 and Q5_41B ne -4)              compute HRACT1= Q5_41B
else if (Q5_40D eq 1 and Q5_41C ne -4)              compute HRACT1= Q5_41C
else if (Q5_40E eq 1 or Q5_40E eq 0)                compute HRACT1= Q5_41D
else                                         compute HRACT1=-3
                                         end if

/* the following lines construct the var HRACT2 */
do if (Q5B2_42 eq -4 or Q5B2_42 eq 0)                 compute HRACT2=-4
else if (Q5B2_43A eq -1 or Q5B2_43B eq -1OR Q5B2_43A eq -2 or Q5B2_43B eq -2)      compute HRACT2=-2
else if (Q5B2_43A eq 995 or Q5B2_43B eq 995)       compute HRACT2=995
else if (Q5B2_43 eq 1)                                compute HRACT2=Q5B2_43A
else if (Q5B2_42 eq 1 and Q5B2_43 eq 0)               compute HRACT2=Q5B2_43B
else if (Q5B_42 eq -4 or Q5B_42 eq 0)                 compute HRACT2=-4
else if (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2)            compute HRACT2=-2
else if (Q5B_43A eq 995 or Q5B_43B eq 995)         compute HRACT2=995
else if (Q5B_43 eq 1)                                compute HRACT2=Q5B_43A
else if (Q5B_42 eq 1 and Q5B_43 eq 0)               compute HRACT2=Q5B_43B
else if (Q5_42 eq -4 or Q5_42 eq 0)                  compute HRACT2=-4
else if (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2)            compute HRACT2=-2
else if (Q5_43A eq 995 or Q5_43B eq 995)         compute HRACT2=995
else if (Q5_43 eq 1)                                compute HRACT2= Q5_43A
else if (Q5_42 eq 1 and Q5_43 eq 0)                compute HRACT2= Q5_43B
else                                         compute HRACT2=-3
                                         end if

```

```

/* the following lines create the var HRACTT */
do if (Q5B2_40A eq -4)
compute HRACTT=-4
else if ((Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
          Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2) and (Q5B2_43A eq -1 or Q5B2_43B eq -1 or
          Q5B2_43A eq -2 or Q5B2_43B eq -2))
compute HRACTT=-2
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and (Q5B2_43A eq
         995 or Q5B2_43B eq 995))
compute HRACTT=995
else if ((Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
          Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2) and (Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=995
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995 or Q5B2_41A eq 995
          or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and (Q5B2_43A eq -1 or Q5B2_43B eq -1
          or Q5B2_43A eq -2 or Q5B2_43B eq -2))
compute HRACTT=995
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41C
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43A eq
          -1 or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41D
else if ((Q5B2_43A ne -4 and Q5B2_43A ne -1 and Q5B2_43A ne -2 and Q5B2_43A ne 995) and (Q5B2_41A eq -1
          or Q5B2_41A eq -2 or Q5B2_41A eq 995 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41B eq 995 or
          Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41C eq 995 or Q5B2_41D eq -1 or Q5B2_41D eq -2 or
          Q5B2_41D eq 995))
compute HRACTT=Q5B2_43A
else if ((Q5B2_43B ne -4 and Q5B2_43B ne -1 and Q5B2_43B ne -2 and Q5B2_43B ne 995) and (Q5B2_41A eq -1
          or Q5B2_41A eq -2 or Q5B2_41A eq 995 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41B eq 995 or
          Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41C eq 995 or Q5B2_41D eq -1 or Q5B2_41D eq -2 or
          Q5B2_41D eq 995))
compute HRACTT=Q5B2_43B
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43A ne -1
          and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41A+Q5B2_43A
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43B ne -1
          and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41A+Q5B2_43B
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43A ne -1
          and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41B+Q5B2_43A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43B ne -1
          and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41B+Q5B2_43B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43A ne -1
          and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41C+Q5B2_43A
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43B ne -1
          and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41C+Q5B2_43B

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43A ne  
-1 and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))  
compute HRACTT=Q5B2_41D+Q5B2_43A  
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43B ne  
-1 and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))  
compute HRACTT=Q5B2_41D+Q5B2_43B  
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and Q5B2_43A eq -4  
and Q5B2_43B eq -4)  
compute HRACTT=Q5B2_41A  
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and Q5B2_43A eq -4  
and Q5B2_43B eq -4)  
compute HRACTT=Q5B2_41B  
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and Q5B2_43A eq -4  
and Q5B2_43B eq -4)  
compute HRACTT=Q5B2_41C  
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and Q5B2_43A eq -4  
and Q5B2_43B eq -4)  
compute HRACTT=Q5B2_41D  
else if ((Q5B2_41A eq -1 or Q5B2_41A eq -2 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41C eq -1 or  
Q5B2_41C eq -2 or Q5B2_41D eq -1 or Q5B2_41D eq -2) and Q5B2_43A eq -4 and Q5B2_43B eq -4)  
compute HRACTT=-2  
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and Q5B2_43A eq -4  
and Q5B2_43B eq -4)  
compute HRACTT=995  
else if (Q5B_40A eq -4)  
compute HRACTT=-4  
else if ((Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq  
-2 or Q5B_41C eq -2 or Q5B_41D eq -2) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or  
Q5B_43B eq -2))  
compute HRACTT=-2  
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq 995 or  
Q5B_43B eq 995))  
compute HRACTT=995  
else if ((Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq  
-2 or Q5B_41C eq -2 or Q5B_41D eq -2) and (Q5B_43A eq 995 or Q5B_43B eq 995))  
compute HRACTT=995  
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995 or Q5B_41A eq 995 or  
Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or  
Q5B_43A eq -2 or Q5B_43B eq -2))  
compute HRACTT=995  
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A eq -1 or  
Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))  
compute HRACTT=Q5B_41A  
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A eq -1 or  
Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))  
compute HRACTT=Q5B_41B  
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A eq -1 or  
Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))  
compute HRACTT=Q5B_41C  
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A eq -1 or  
Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))  
compute HRACTT=Q5B_41D  
else if ((Q5B_43A ne -4 and Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995) and (Q5B_41A eq -1 or  
Q5B_41A eq -2 or Q5B_41A eq 995 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41B eq 995 or Q5B_41C  
eq -1 or Q5B_41C eq -2 or Q5B_41C eq 995 or Q5B_41D eq -1 or Q5B_41D eq -2 or Q5B_41D eq 995))  
compute HRACTT=Q5B_43A
```

```

else if ((Q5B_43B ne -4 and Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995) and (Q5B_41A eq -1 or
        Q5B_41A eq -2 or Q5B_41A eq 995 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41B eq 995 or Q5B_41C
        eq -1 or Q5B_41C eq -2 or Q5B_41C eq 995 or Q5B_41D eq -1 or Q5B_41D eq -2 or Q5B_41D eq 995))
compute HRACTT=Q5B_43B
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41A+Q5B_43A
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41A+Q5B_43B
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41B+Q5B_43A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41B+Q5B_43B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41C+Q5B_43A
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41C+Q5B_43B
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A ne -1 and
        Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41D+Q5B_43A
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43B ne -1 and
        Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41D+Q5B_43B
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41C
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=Q5B_41D
else if ((Q5B_41A eq -1 or Q5B_41A eq -2 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41C eq -1 or Q5B_41C eq
        -2 or Q5B_41D eq -1 or Q5B_41D eq -2) and Q5B_43A eq -4 and Q5B_43B eq -4)
compute HRACTT=-2
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and Q5B_43A eq -4 and
        Q5B_43B eq -4)
compute HRACTT=995
else if (R(52404.) eq -4)
compute HRACTT=-4
else if ((Q5_41A eq -1 or Q5_41B eq -1 or Q5_41C eq -1 or Q5_41D eq -1 or Q5_41A eq -2 or Q5_41B eq -2 or
        Q5_41C eq -2 or Q5_41D eq -2) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2))
compute HRACTT=-2
else if ((Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995) and (Q5_43A eq 995 or Q5_43B
        eq 995))
compute HRACTT=995
else if ((Q5_41A eq -1 or Q5_41B eq -1 or Q5_41C eq -1 or Q5_41D eq -1 or Q5_41A eq -2 or Q5_41B eq -2 or
        Q5_41C eq -2 or Q5_41D eq -2) and (Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=995

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

else if ((Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995 or Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2))
compute HRACTT=995
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT= Q5_41A
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT= Q5_41B
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT= Q5_41C
else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT= Q5_41D
else if ((Q5_43A ne -4 and Q5_43A ne -1 and Q5_43A ne -2 and Q5_43A ne 995) and (Q5_41A eq -1 or Q5_41A eq -2 or Q5_41A eq 995 or Q5_41B eq -1 or Q5_41B eq -2 or Q5_41B eq 995 or Q5_41C eq -1 or Q5_41C eq -2 or Q5_41C eq 995 or Q5_41D eq -1 or Q5_41D eq -2 or Q5_41D eq 995))
compute HRACTT= Q5_43A
else if ((Q5_43B ne -4 and Q5_43B ne -1 and Q5_43B ne -2 and Q5_43B ne 995) and Q5_41A eq -1 or Q5_41A eq -2 or Q5_41A eq 995 or Q5_41B eq -1 or Q5_41B eq -2 or Q5_41B eq 995 or Q5_41C eq -1 or Q5_41C eq -2 or Q5_41C eq 995 or Q5_41D eq -1 or Q5_41D eq -2 or Q5_41D eq 995))
compute HRACTT= Q5_43B
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and (Q5_43A ne -1 and Q5_43A ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT= Q5_41A+Q5_43A
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and (Q5_43B ne -1 and Q5_43B ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT= Q5_41A+Q5_43B
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and (Q5_43A ne -1 and Q5_43A ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT= Q5_41B+ Q5_43A
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and (Q5_43B ne -1 and Q5_43B ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT= Q5_41B+ Q5_43B
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and (Q5_43A ne -1 and Q5_43A ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT= Q5_41C+ Q5_43A
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and (Q5_43B ne -1 and Q5_43B ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT= Q5_41C Q5_43B
else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and (Q5_43A ne -1 and Q5_43A ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT= Q5_41D+ Q5_43A
else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and (Q5_43B ne -1 and Q5_43B ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT= Q5_41D+ Q5_43B
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and Q5_43A eq -4 and Q5_43B eq -4)
compute HRACTT= Q5_41A
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and Q5_43A eq -4 and Q5_43B eq -4)
compute HRACTT= Q5_41B
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and Q5_43A eq -4 and Q5_43B eq -4)
compute HRACTT= Q5_41C

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and Q5_43A eq -4 and Q5_43B
        eq -4)
compute HRACTT= Q5_41D
else if ((Q5_41A eq -1 or Q5_41A eq -2 or Q5_41B eq -1 or Q5_41B eq -2 or Q5_41C eq -1 or Q5_41C eq -2 or
        Q5_41D eq -1 or Q5_41D eq -2) and Q5_43A eq -4 and Q5_43B eq -4)
compute HRACTT=-2
else if ((Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995) and Q5_43A eq -4 and Q5_43B
        eq -4)
compute HRACTT=995
else      compute HRACTT=-3                                end if

/* the following lines creates the var BUS2 */
do if (R(52322.) eq -4)                               compute BUS2=-4
else if (R(52322.) eq -1)                             compute BUS2=-1
else if (R(52322.) eq -2)                             compute BUS2=-2
else if (R(52323.) eq -1 or R(52324.) eq -1)         compute BUS2=-1
else if (R(52323.) eq -2 or R(52324.) eq -2)         compute BUS2=-2
else if (R(52323.) eq 1 or R(52324.) eq 1)           compute BUS2=1
else if (R(52324.) eq 0)                             compute BUS2=2 /* this should normally be 0 */
else                                         compute BUS2=-3                                end if

/* the following lines create the var HRUSL1 */
do if (Q5BCK2_25 eq -4)                               compute HRUSL1=-4
else if (Q5BCK2_26A eq -1 or Q5BCK2_26C eq -1 or Q5BCK2_26A eq -2 or Q5BCK2_26C eq -2)
        compute HRUSL1=-2
else if (Q5BCK2_26A eq 995 or Q5BCK2_26C eq 995)   compute HRUSL1=995
else if (Q5BCK2_26A eq 1)                            compute HRUSL1=Q5B2_26B
else if (Q5BCK2_26C eq 1)                            compute HRUSL1=Q5B2_26D
else if (R(52486.) eq -4)                           compute HRUSL1=-4
else if (Q5B_26A eq -1 or R(52487.) eq -1 or Q5B_26A eq -2 or R(52487.) eq -2)
        compute HRUSL1=-2
else if (Q5B_26A eq 995 or R(52487.) eq 995)       compute HRUSL1=995
else if (Q5B_26A eq 1)                             compute HRUSL1=Q5B_26B
else if (R(52487.) eq 1)                           compute HRUSL1=R(52488.)
else if (R(52360.) eq -4)                           compute HRUSL1=-4
else if (R(52361.) eq -1 or R(52363.) eq -1 or R(52361.) eq -2 or R(52363.) eq -2)
        compute HRUSL1=-2
else if (R(52361.) eq 995 or R(52363.) eq 995)    compute HRUSL1=995
else if (R(52361.) eq 1)                           compute HRUSL1=R(52362.)
else if (R(52363.) eq 1)                           compute HRUSL1=R(52364.)
else                                         compute HRUSL1=-3                                end if

/* the following lines create the var HRUSL2 */
do if (Q5B2_28A eq -4)                               compute HRUSL2=-4
else if (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B eq -2 or Q5B2_28D eq -2)   compute HRUSL2=-2
else if (Q5B2_28B eq 995 or Q5B2_28D eq 995)     compute HRUSL2=995
else if (Q5B2_28B eq 1)                            compute HRUSL2=Q5B2_28C
else if (Q5B2_28D eq 1)                            compute HRUSL2=Q5B2_28E
else if (Q5B_28A eq -4)                           compute HRUSL2=-4
else if (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2)        compute HRUSL2=-2
else if (Q5B_28B eq 995 or Q5B_28D eq 995)       compute HRUSL2=995
else if (Q5B_28B eq 1)                           compute HRUSL2=Q5B_28C
else if (Q5B_28D eq 1)                           compute HRUSL2=Q5B_28E
else if (R(52366.) eq -4)                           compute HRUSL2=-4
else if (R(52367.) eq -1 or R(52369.) eq -1 or R(52367.) eq -2 or R(52369.) eq -2)  compute HRUSL2=-2
else if (R(52367.) eq 995 or R(52369.) eq 995)  compute HRUSL2=995
else if (R(52367.) eq 1)                           compute HRUSL2=Q5_28C

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
else if (R(52369.) eq 1)           compute HRUSL2=Q5_28E
else                                compute HRUSL2=-3               end if

/* the following lines create the var HRUSLT */
do if (Q5BCK2_25 eq -4)
compute HRUSLT=-4
else if ((Q5BCK2_26A eq -1 or Q5BCK2_26C eq -1 or Q5BCK2_26A eq -2 or Q5BCK2_26C eq -2) and
         (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B eq -2 or Q5B2_28D eq -2))
compute HRUSLT=-2
else if ((Q5BCK2_26A eq 995 or Q5BCK2_26C eq 995) and (Q5B2_28B eq 995 or Q5B2_28D eq 995))
compute HRUSLT=995
else if ((Q5BCK2_26A eq -1 or Q5BCK2_26C eq -1 or Q5BCK2_26A eq -2 or Q5BCK2_26C eq -2) and
         (Q5B2_28B eq 995 or Q5B2_28D eq 995))
compute HRUSLT=995
else if ((Q5BCK2_26A eq 995 or Q5BCK2_26C eq 995) and (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B
eq -2 or Q5B2_28D eq -2))
compute HRUSLT=995
else if ((Q5BCK2_26A eq 995 or Q5BCK2_26C eq 995) and Q5B2_27 ne 1)
compute HRUSLT=995
else if ((Q5BCK2_26A eq -1 or Q5BCK2_26C eq -1 or Q5BCK2_26A eq -2 or Q5BCK2_26C eq -2) and Q5B2_27
ne 1)
compute HRUSLT=-2
else if (Q5BCK2_26A eq 1 and (Q5B2_28B eq -1 or Q5B2_28B eq -2 or Q5B2_28B eq 995 or Q5B2_28D eq -1 or
      Q5B2_28D eq -2 or Q5B2_28D eq 995))
compute HRUSLT=Q5B2_26B
else if (Q5BCK2_26C eq 1 and (Q5B2_28B eq -1 or Q5B2_28B eq -2 or Q5B2_28B eq 995 or Q5B2_28D eq -1 or
      Q5B2_28D eq -2 or Q5B2_28D eq 995))
compute HRUSLT=Q5B2_26D
else if (Q5B2_28B eq 1 and (Q5BCK2_26A eq -1 or Q5BCK2_26A eq -2 or Q5BCK2_26A eq 995 or Q5BCK2_26C
eq -1 or Q5BCK2_26C eq -2 or Q5BCK2_26C eq 995))
compute HRUSLT=Q5B2_28C
else if (Q5B2_28D eq 1 and (Q5BCK2_26A eq -1 or Q5BCK2_26A eq -2 or Q5BCK2_26A eq 995 or
      Q5BCK2_26C eq -1 or Q5BCK2_26C eq -2 or Q5BCK2_26C eq 995))
compute HRUSLT=Q5B2_28E
else if (Q5BCK2_26A eq 1 and Q5B2_27 ne 1)           compute HRUSLT=Q5B2_26B
else if (Q5BCK2_26C eq 1 and Q5B2_27 ne 1)           compute HRUSLT=Q5B2_26D
else if (Q5BCK2_26A eq 1 and Q5B2_28B eq 1)          compute HRUSLT=Q5B2_26B+Q5B2_28C
else if (Q5BCK2_26A eq 1 and Q5B2_28D eq 1)          compute HRUSLT=Q5B2_26B+Q5B2_28E
else if (Q5BCK2_26C eq 1 and Q5B2_28B eq 1)          compute HRUSLT=Q5B2_26D+Q5B2_28C
else if (Q5BCK2_26C eq 1 and Q5B2_28D eq 1)          compute HRUSLT= Q5BCK2_26C+Q5B2_28E
else if (R(52486.) eq -4)                            compute HRUSLT=-4
else if ((Q5B_26A eq -1 or R(52487.) eq -1 or Q5B_26A eq -2 or R(52487.) eq -2) and (Q5B_28B eq -1 or Q5B_28D
eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2))        compute HRUSLT=-2
else if ((Q5B_26A eq 995 or R(52487.) eq 995) and (Q5B_28B eq 995 or Q5B_28D eq 995))        compute HRUSLT=995
else if ((Q5B_26A eq -1 or R(52487.) eq -1 or Q5B_26A eq -2 or R(52489.) eq -2) and (Q5B_28B eq 995 or
      Q5B_28D eq 995))                            compute HRUSLT=995
else if ((Q5B_26A eq 995 or R(52487.) eq 995) and (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or
      Q5B_28D eq -2))                            compute HRUSLT=995
else if ((Q5B_26A eq 995 or R(52487.) eq 995) and R(52489.) ne 1)                          compute HRUSLT=995
else if ((Q5B_26A eq -1 or R(52487.) eq -1 or Q5B_26A eq -2 or R(52487.) eq -2) and R(52489.) ne 1)
compute HRUSLT=-2
```

```

else if (Q5B_26A eq 1 and (Q5B_28B eq -1 or Q5B_28B eq -2 or Q5B_28B eq 995 or Q5B_28D eq -1 or Q5B_28D
eq -2 or Q5B_28D eq 995))
compute HRUSLT=Q5B_26B
else if (R(52487.) eq 1 and (Q5B_28B eq -1 or Q5B_28B eq -2 or Q5B_28B eq 995 or Q5B_28D eq -1 or Q5B_28D
eq -2 or Q5B_28D eq 995))
compute HRUSLT=R(52488.)
else if (Q5B_28B eq 1 and (Q5B_26A eq -1 or Q5B_26A eq -2 or Q5B_26A eq 995 or R(52487.) eq -1 or R(52487.)
eq -2 or R(52487.) eq 995))
compute HRUSLT=Q5B_28C
else if (Q5B_28D eq 1 and (Q5B_26A eq -1 or Q5B_26A eq -2 or Q5B_26A eq 995 or R(52487.) eq -1 or R(52487.)
eq -2 or R(52487.) eq 995))
compute HRUSLT=Q5B_28E
else if (Q5B_26A eq 1 and R(52489.) ne 1)                               compute HRUSLT=Q5B_26B
else if (R(52487.) eq 1 and R(52489.) ne 1)                             compute HRUSLT=R(52488.)
else if (Q5B_26A eq 1 and Q5B_28B eq 1)                                compute HRUSLT=Q5B_26B+Q5B_28C
else if (Q5B_26A eq 1 and Q5B_28D eq 1)                                compute HRUSLT=Q5B_26B+Q5B_28E
else if (R(52487.) eq 1 and Q5B_28B eq 1)                                compute HRUSLT=R(52488.)+Q5B_28C
else if (R(52487.) eq 1 and Q5B_28D eq 1)                                compute HRUSLT=R(52487.)+Q5B_28E
else if (R(52360.) eq -4)                                                 compute HRUSLT=-4
else if ((R(52361.) eq -1 or R(52363.) eq -1 or R(52361.) eq -2 or R(52363.) eq -2) and (R(52367.) eq -1 or R(52369.)
eq -1 or R(52367.) eq -2 or R(52369.) eq -2))
compute HRUSLT=-2
else if ((R(52361.) eq 995 or R(52363.) eq 995) and (R(52367.) eq 995 or R(52369.) eq 995))
compute HRUSLT=995
else if ((R(52361.) eq -1 or R(52363.) eq -1 or R(52361.) eq -2 or R(52363.) eq -2) and (R(52367.) eq 995 or
R(52369.) eq 995))
compute HRUSLT=995
else if ((R(52361.) eq 995 or R(52363.) eq 995) and (R(52367.) eq -1 or R(52369.) eq -1 or R(52367.) eq -2 or
R(52369.) eq -2))
compute HRUSLT=995
else if ((R(52361.) eq 995 or R(52363.) eq 995) and R(52365.) ne 1)
compute HRUSLT=995
else if ((R(52361.) eq -1 or R(52363.) eq -1 or R(52361.) eq -2 or R(52363.) eq -2) and R(52365.) ne 1)
compute HRUSLT=-2
else if (R(52361.) eq 1 and (R(52367.) eq -1 or R(52367.) eq -2 or R(52367.) eq 995 or R(52369.) eq -1 or R(52369.)
eq -2 or R(52369.) eq 995))
compute HRUSLT=R(52362.)
else if (R(52363.) eq 1 and (R(52367.) eq -1 or R(52367.) eq -2 or R(52367.) eq 995 or R(52369.) eq -1 or R(52369.)
eq -2 or R(52369.) eq 995))
compute HRUSLT=R(52364.)
else if (R(52367.) eq 1 and (R(52361.) eq -1 or R(52361.) eq -2 or R(52361.) eq 995 or R(52363.) eq -1 or R(52363.)
eq -2 or R(52363.) eq 995))
compute HRUSLT=R(52368.)
else if (R(52369.) eq 1 and (R(52361.) eq -1 or R(52361.) eq -2 or R(52361.) eq 995 or R(52363.) eq -1 or R(52363.)
eq -2 or R(52363.) eq 995))
compute HRUSLT=R(52370.)
else if (R(52361.) eq 1 and R(52365.) ne 1)                               compute HRUSLT=R(52362.)
else if (R(52363.) eq 1 and R(52365.) ne 1)                             compute HRUSLT=R(52364.)
else if (R(52361.) eq 1 and R(52367.) eq 1)                                compute HRUSLT=R(52362.)+R(52368.)
else if (R(52361.) eq 1 and R(52369.) eq 1)                                compute HRUSLT=R(52362.)+R(52370.)
else if (R(52363.) eq 1 and R(52367.) eq 1)                                compute HRUSLT=R(52364.)+R(52368.)
else if (R(52363.) eq 1 and R(52369.) eq 1)                                compute HRUSLT=R(52363.)+R(52370.)
else                                                               compute HRUSLT=-3                                         end if

/* the following lines create the var HRFTPT */
do if (Q5B2_30C eq -4)                                                 compute HRFTPT=-4
else if (Q5B2_30F eq -1 or Q5B2_30G eq -1 or Q5B2_30H eq -1 or Q5B2_30I eq -1)      compute HRFTPT=-1

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

else if (Q5B2_30F eq -2 or Q5B2_30G eq -2 or Q5B2_30H eq -2 or Q5B2_30I eq -2)      compute HRFTPT=-2
else if (Q5B2_30F ne -4)                  compute HRFTPT=Q5B2_30F
else if (Q5B2_30G ne -4)                  compute HRFTPT=Q5B2_30G
else if (Q5B2_30H ne -4)                  compute HRFTPT=Q5B2_30H
else if (Q5B2_30I ne -4)                  compute HRFTPT=Q5B2_30I
else if (Q5B_30C eq -4)                  compute HRFTPT=-4
else if (Q5B_30F eq -1 or Q5B_30G eq -1 or Q5B_30H eq -1 or Q5B_30I eq -1)      compute HRFTPT=-1
else if (Q5B_30F eq -2 or Q5B_30G eq -2 or Q5B_30H eq -2 or Q5B_30I eq -2)      compute HRFTPT=-2
else if (Q5B_30F ne -4)                  compute HRFTPT=Q5B_30F
else if (Q5B_30G ne -4)                  compute HRFTPT=Q5B_30G
else if (Q5B_30H ne -4)                  compute HRFTPT=Q5B_30H
else if (Q5B_30I ne -4)                  compute HRFTPT=Q5B_30I
else if (R(52381.) eq -4)                compute HRFTPT=-4
else if (Q5_30F eq -1 or R(52384.) eq -1 or R(52385.) eq -1 or R(52386.) eq -1)    compute HRFTPT=-1
else if (Q5_30F eq -2 or R(52384.) eq -2 or R(52385.) eq -2 or R(52386.) eq -2)    compute HRFTPT=-2
else if (Q5_30F ne -4)                  compute HRFTPT=Q5_30F
else if (R(52384.) ne -4)                compute HRFTPT=R(52384.)
else if (R(52385.) ne -4)                compute HRFTPT=R(52385.)
else if (R(52386.) ne -4)                compute HRFTPT=R(52386.)
else                                         compute HRFTPT=-3                                end if

/* the following lines create the var HRWANT */
do if (Q5B2_33 ne -4)                  compute HRWANT=Q5B2_33
else if (Q5B_33 ne -4)                 compute HRWANT=Q5B_33
else                                         compute HRWANT=R(52397.)                                end if

/* the following lines create the var HRRSN1 */
do if (Q5B2_34 ne -4)                  compute HRRSN1=Q5B2_34
else if (Q5B_34 ne -4)                 compute HRRSN1=Q5B_34
else                                         compute HRRSN1=R(52398.)                                end if

/* the following lines create the var HRCK7 */
compute HRCK7=-4
do if ((BUS2 eq 2 or BUS2 eq -2 or BUS2 eq -1) and (HRACT1 LT 15 or HRACT1 eq -2))   compute HRCK7=1
compute HRCK7=1
else if ((BUS2 eq 2 or BUS2 eq -2 or BUS2 eq -1) and HRACT1 GE 15)                   compute HRCK7=2
compute HRCK7=2
else if ((HRUSLT GE 35 or HRFTPT eq 1) and HRACTT LT 35 and (HRACT1 ne -1 or HRACT1 ne -2 or
          HRACT2 ne -1 or HRACT2 ne -2))                                         compute HRCK7=3
compute HRCK7=3
else if (HRWANT eq 1 and HRACTT LT 35 and (HRRSN1 eq 1 or HRRSN1 eq 2 or HRRSN1 eq 3))  compute HRCK7=4
compute HRCK7=4
else                                         compute HRCK7=5                                end if

/* the following lines create the var ABSOT */
do if (HRACT1 eq 0 and HRACT2 eq 0)    compute ABSOT=1
else                                         compute ABSOT=0                                end if

/* the following lines create the var HRCK6 */
compute HRCK6=-4
do if ((HRACT1 eq 0 and HRACT2 eq 0) and (BUS2 eq 2 or BUS2 eq -1 or BUS2 eq -2))   compute HRCK6=1
compute HRCK6=1
else if (ABSOT eq 1)                  compute HRCK6=2
else                                         compute HRCK6=3                                end if

/* the following lines create the var WK */
compute WK=-4

```

```

do if (R(52318.) ne -4)           compute WK=R(52318.)
else if (R(52319.) ne -4)         compute WK=R(52319.)           end if

/* the following lines create the var RETOT */
do if (WK eq 3)                  compute RETOT=1
else                           compute RETOT=0           end if

/* the following lines rename the vars */
compute BUS1=R(52321.)          compute RET1=R(52335.)
compute IO1COW=R(52314.)        compute NLFACT=R(52578.)

/* the following lines create additional variables */
do if (Q5B2_22 ne -4)           compute ABSPD=Q5B2_22
else if (R(52481.) ne -4)       compute ABSPD=R(52481.)
else                           compute ABSPD=R(52354.)           end if

do if (Q5B2_65 ne -4)           compute LKAVL=Q5B2_65
else if (R(52515.) ne -4)       compute LKAVL=R(52515.)
else                           compute LKAVL=R(52437.)           end if

do if (Q5B2_51 ne -4)           compute LAYAVL=Q5B2_51
else if (Q5B_51 ne -4)          compute LAYAVL=Q5B_51
else                           compute LAYAVL=R(52410.)           end if

do if (Q5B2_66 ne -4)           compute LKAVR=Q5B2_66
else if (Q5B_66 ne -4)          compute LKAVR=Q5B_66
else                           compute LKAVR=R(45540.)           end if

do if (Q5B2_52 ne -4)           compute LAYAVR=Q5B2_52
else if (Q5B_52 ne -4)          compute LAYAVR=Q5B_52
else                           compute LAYAVR=R(52411.)           end if

do if (Q5B2_60 ne -4)           compute LK=Q5B2_60
else if (R(52503.) ne -4)       compute LK=R(52503.)
else                           compute LK=R(52420.)           end if

do if (Q5B2_11 ne -4)           compute DIS1=Q5B2_11
else if (R(52455.) ne -4)       compute DIS1=R(52455.)
else                           compute DIS1=R(52336.)           end if

do if (Q5B2_12 ne -4)           compute DIS2=Q5B2_12
else if (R(52456.) ne -4)       compute DIS2=R(52456.)
else                           compute DIS2=R(52337.)           end if

/* the following lines create the var DWWNTO */
do if (RET1 eq 1)                compute DWWNTO=1
else if (Q5B2_78 eq 1)           compute DWWNTO=1
else if (Q5B2_78 eq 0)           compute DWWNTO=2
else if (Q5B2_78 eq -2)          compute DWWNTO=-2
else if (Q5B2_78 eq -1)          compute DWWNTO=-1
else if (Q5B2_78 eq 3)           compute DWWNTO=3
else if (Q5B2_78 eq 4)           compute DWWNTO=4
else if (Q5BCK_78 eq 1)          compute DWWNTO=1
else if (Q5BCK_78 eq 0)          compute DWWNTO=2
else if (Q5BCK_78 eq -2)         compute DWWNTO=-2
else if (Q5BCK_78 eq -1)         compute DWWNTO=-1
else if (Q5BCK_78 eq 3)          compute DWWNTO=3

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

else if (Q5BCK_78 eq 4)      compute DWWNTO=4
else if (Q5_78 eq 1)         compute DWWNTO=1
else if (Q5_78 eq 0)         compute DWWNTO=2
else if (Q5_78 eq -2)        compute DWWNTO=-2
else if (Q5_78 eq -1)        compute DWWNTO=-1
else if (Q5_78 eq 3)         compute DWWNTO=3
else if (Q5_78 eq 4)         compute DWWNTO=4
else                         compute DWWNTO=-3           end if

/* the following lines create the var DIS */
do if (Q5_9A4 eq -4)
compute DIS=-4
else if (Q5_9D eq -1 or Q5_9E eq -1 or Q5_9F eq -1 or Q5_9G eq -1) or (Q5_9D eq -2 or Q5_9E eq -2 or Q5_9F eq -2 or Q5_9G eq -2)
compute DIS=-2
else if (Q5_9A4 eq 1 and Q5_9D eq 1)      compute DIS=1
else if (Q5_9D eq 0)                      compute DIS=0
else if (Q5_9A5 eq 1 and Q5_9E eq 1)      compute DIS=1
else if (Q5_9E eq 0)                      compute DIS=0
else if (Q5_9A6 eq 1 and Q5_9F eq 1)      compute DIS=1
else if (Q5_9F eq 0)                      compute DIS=0
else if (Q5_9A7 eq 1 and Q5_9G eq 1)      compute DIS=1
else if (Q5_9G eq 0)                      compute DIS=0
else                                         compute DIS=-3           end if

/* the following lines create IO1COW */
do if (R(52314.) eq 1)      compute IO1COW=8
else                         compute IO1COW=0           end if

/* the following lines create the ESR var approximating MLR */
do if ((WK eq 1 and (HRCK6 eq 3 or HRCK6 eq -4 or HRCK6 eq 99 or HRCK6 eq 999)) or (BUS1 eq 1 and
(HRCK7 eq 2 or HRCK7 eq 3 or HRCK7 eq 4 or HRCK7 eq 5 or HRCK7 eq -4)))
compute ESR=1
else if (ABSPD eq 1 or ABSPD eq 0 or ABSPD eq -1 or ABSPD eq -2)
compute ESR=2
else if ((LAYAVL eq 1 or LAYAVL eq -1 or LAYAVL eq -2) or LAYAVR eq 1)
compute ESR=3
else if ((LKAVL eq 1 or LKAVL eq -1 or LKAVL eq -2) or (LKAVR eq 1 or LKAVR eq 2))
compute ESR=4
else if ((WK eq 3 and AGE GE 50 and RET1 eq 1 and LK eq 1 and LKAVL eq 0 and (LKAVR eq 3 or LKAVR eq 4 or
LKAVR eq -1 or LKAVR eq -2)) or (RET1 eq 0 or LK eq 3 or DWWNTO eq 3) or NLFAC1T eq 5 or
RETOT eq 1)
compute ESR=5
else if (DIS1 eq 1 or DIS2 eq 1 or DIS eq 1)      compute ESR=6
else if (R(52314.) eq 1)                          compute ESR=8
else                                         compute ESR=7           end if

ESR(6)=R(15199.);          ESR(7)=R(18906.);          ESR(8)=R(22577.);
ESR(9)=R(24451.);          ESR(10)=R(28706.);         ESR(11)=R(30743.);
ESR(12)=R(34010.);          ESR(13)=R(36564.);         ESR(14)=R(40069.);
ESR(15)=R(44180.);          ESR(16)=R(50810.);         ESR(17)=R(51663.);
ESRC(6)=R(15199.01);       ESRC(7)=R(18906.01);        ESRC(8)=R(22577.01);
ESRC(9)=R(24451.01);       ESRC(10)=R(28707.);        ESRC(11)=R(30744.);
ESRC(12)=R(34011.);        ESRC(13)=R(36565.);        ESRC(14)=R(40070.);
ESRC(15)=R(44181.);        ESRC(16)=R(50811.);        ESRC(17)=R(51664.);


```

```

/* 1998 VARIABLES */

/* The programming for the new 1998 ESR variables is done in SPSS */
/* the following lines create the var AGE */
compute AGE=41

/* the following lines construct the var HRACT1 */
do if (Q5B2_40A eq -4)
compute HRACT1=-4
else if (Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
        Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2)
compute HRACT1=-2
else if (Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995)
compute HRACT1=995
else if (Q5B2_40B eq 1 and Q5B2_41A ne -4)           compute HRACT1=Q5B2_41A
else if (Q5B2_40C eq 1 and Q5B2_41B ne -4)           compute HRACT1=Q5B2_41B
else if (Q5B2_40D eq 1 and Q5B2_41C ne -4)           compute HRACT1=Q5B2_41C
else if (Q5B2_40E eq 1 or Q5B2_40E eq 0)             compute HRACT1=Q5B2_41D
else if (Q5B_40A eq -4)                               compute HRACT1=-4
else if (Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq
        -2 or Q5B_41C eq -2 or Q5B_41D eq -2)
compute HRACT1=-2
else if (Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995)
compute HRACT1=995
else if (Q5B_40B eq 1 and Q5B_41A ne -4)           compute HRACT1=Q5B_41A
else if (Q5B_40C eq 1 and Q5B_41B ne -4)           compute HRACT1=Q5B_41B
else if (Q5B_40D eq 1 and Q5B_41C ne -4)           compute HRACT1=Q5B_41C
else if (Q5B_40E eq 1 or Q5B_40E eq 0)             compute HRACT1=Q5B_41D
else if (Q5_40A eq -4)                             compute HRACT1=-4
else if (Q5_41A eq -1 or Q5_41B eq -1 or Q5_41C eq -1 or Q5_41D eq -1 or Q5_41A eq -2 or Q5_41B eq -2 or
        Q5_41C eq -2 or Q5_41D eq -2)
compute HRACT1=-2
else if (Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995)
compute HRACT1=995
else if (Q5_40B eq 1 and Q5_41A ne -4)           compute HRACT1=Q5_41A
else if (Q5_40C eq 1 and Q5_41B ne -4)           compute HRACT1=Q5_41B
else if (Q5_40D eq 1 and Q5_41C ne -4)           compute HRACT1=Q5_41C
else if (Q5_40E eq 1 or Q5_40E eq 0)             compute HRACT1=Q5_41D
else if (Q5_40A eq -4)                           compute HRACT1=-3
else                                         end if

/* the following lines construct the var HRACT2 */
do if (Q5B2_42 eq -4 or Q5B2_42 eq 0)               compute HRACT2=-4
else if (Q5B2_43A eq -1 or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2)      compute HRACT2=-2
else if (Q5B2_43A eq 995 or Q5B2_43B eq 995)       compute HRACT2=995
else if (Q5B2_43 eq 1)                            compute HRACT2=Q5B2_43A
else if (Q5B2_42 eq 1 and Q5B2_43 eq 0)           compute HRACT2=Q5B2_43B
else if (Q5B_42 eq -4 or Q5B_42 eq 0)             compute HRACT2=-4
else if (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2)      compute HRACT2=-2
else if (Q5B_43A eq 995 or Q5B_43B eq 995)       compute HRACT2=995
else if (Q5B_43 eq 1)                            compute HRACT2=Q5B_43A
else if (Q5B_42 eq 1 and Q5B_43 eq 0)           compute HRACT2=Q5B_43B
else if (Q5_42 eq -4 or Q5_42 eq 0)              compute HRACT2=-4
else if (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2)      compute HRACT2=-2
else if (Q5_43A eq 995 or Q5_43B eq 995)       compute HRACT2=995
else if (Q5_43 eq 1)                            compute HRACT2=Q5_43A
else if (Q5_42 eq 1 and Q5_43 eq 0)           compute HRACT2=Q5_43B
else                                         compute HRACT2=-3
end if

```

```

/* the following lines create the var HRACTT */
do if (Q5B2_40A eq -4)
compute HRACTT=-4
else if ((Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
          Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2) and (Q5B2_43A eq -1 or Q5B2_43B eq -1 or
          Q5B2_43A eq -2 or Q5B2_43B eq -2))
compute HRACTT=-2
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and (Q5B2_43A eq
         995 or Q5B2_43B eq 995))
compute HRACTT=995
else if ((Q5B2_41A eq -1 or Q5B2_41B eq -1 or Q5B2_41C eq -1 or Q5B2_41D eq -1 or Q5B2_41A eq -2 or
          Q5B2_41B eq -2 or Q5B2_41C eq -2 or Q5B2_41D eq -2) and (Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=995
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995 or Q5B2_41A eq 995
          or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and (Q5B2_43A eq -1 or Q5B2_43B eq -1
          or Q5B2_43A eq -2 or Q5B2_43B eq -2))
compute HRACTT=995
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41C
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43A eq -1
          or Q5B2_43B eq -1 or Q5B2_43A eq -2 or Q5B2_43B eq -2 or Q5B2_43A eq 995 or Q5B2_43B eq 995))
compute HRACTT=Q5B2_41D
else if ((Q5B2_43B ne -4 and Q5B2_43A ne -1 and Q5B2_43A ne -2 and Q5B2_43A ne 995) and (Q5B2_41A eq -1
          or Q5B2_41A eq -2 or Q5B2_41A eq 995 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41B eq 995 or
          Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41C eq 995 or Q5B2_41D eq -1 or Q5B2_41D eq -2 or
          Q5B2_41D eq 995))
compute HRACTT=Q5B2_43A
else if ((Q5B2_43A ne -4 and Q5B2_43B ne -1 and Q5B2_43B ne -2 and Q5B2_43B ne 995) and (Q5B2_41A eq -1
          or Q5B2_41A eq -2 or Q5B2_41A eq 995 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41B eq 995 or
          Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41C eq 995 or Q5B2_41D eq -1 or Q5B2_41D eq -2 or
          Q5B2_41D eq 995))
compute HRACTT=Q5B2_43B
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43A ne -1
          and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41A+Q5B2_43A
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and (Q5B2_43B ne -1
          and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41A+Q5B2_43B
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43A ne -1
          and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41B+Q5B2_43A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and (Q5B2_43B ne -1
          and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41B+Q5B2_43B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43A ne -1
          and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41C+Q5B2_43A
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and (Q5B2_43B ne -1
          and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41C+Q5B2_43B

```

```

else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43A ne -1 and Q5B2_43A ne -2 and Q5B2_43A ne 995 and Q5B2_43A ne -4))
compute HRACTT=Q5B2_41D+Q5B2_43A
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and (Q5B2_43B ne -1 and Q5B2_43B ne -2 and Q5B2_43B ne 995 and Q5B2_43B ne -4))
compute HRACTT=Q5B2_41D+Q5B2_43B
else if ((Q5B2_41A ne -4 and Q5B2_41A ne -1 and Q5B2_41A ne -2 and Q5B2_41A ne 995) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41A
else if ((Q5B2_41B ne -4 and Q5B2_41B ne -1 and Q5B2_41B ne -2 and Q5B2_41B ne 995) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41B
else if ((Q5B2_41C ne -4 and Q5B2_41C ne -1 and Q5B2_41C ne -2 and Q5B2_41C ne 995) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41C
else if ((Q5B2_41D ne -4 and Q5B2_41D ne -1 and Q5B2_41D ne -2 and Q5B2_41D ne 995) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=Q5B2_41D
else if ((Q5B2_41A eq -1 or Q5B2_41A eq -2 or Q5B2_41B eq -1 or Q5B2_41B eq -2 or Q5B2_41C eq -1 or Q5B2_41C eq -2 or Q5B2_41D eq -1 or Q5B2_41D eq -2) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=-2
else if ((Q5B2_41A eq 995 or Q5B2_41B eq 995 or Q5B2_41C eq 995 or Q5B2_41D eq 995) and Q5B2_43A eq -4 and Q5B2_43B eq -4)
compute HRACTT=995

else if (Q5B_40A eq -4)
compute HRACTT=-4
else if ((Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq -2 or Q5B_41C eq -2 or Q5B_41D eq -2) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2))
compute HRACTT=-2
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=995
else if ((Q5B_41A eq -1 or Q5B_41B eq -1 or Q5B_41C eq -1 or Q5B_41D eq -1 or Q5B_41A eq -2 or Q5B_41B eq -2 or Q5B_41C eq -2 or Q5B_41D eq -2) and (Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=995
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995 or Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2))
compute HRACTT=995
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41C
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A eq -1 or Q5B_43B eq -1 or Q5B_43A eq -2 or Q5B_43B eq -2 or Q5B_43A eq 995 or Q5B_43B eq 995))
compute HRACTT=Q5B_41D
else if ((Q5B_43B ne -4 and Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995) and (Q5B_41A eq -1 or Q5B_41A eq -2 or Q5B_41A eq 995 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41B eq 995 or Q5B_41C eq -1 or Q5B_41C eq -2 or Q5B_41C eq 995 or Q5B_41D eq -1 or Q5B_41D eq -2 or Q5B_41D eq 995))
compute HRACTT=Q5B_43A

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

else if ((Q5B_43A ne -4 and Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995) and (Q5B_41A eq -1 or Q5B_41A eq -2 or Q5B_41A eq 995 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41B eq 995 or Q5B_41C eq -1 or Q5B_41C eq -2 or Q5B_41C eq 995 or Q5B_41D eq -1 or Q5B_41D eq -2 or Q5B_41D eq 995))
compute HRACTT=Q5B_43B
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41A+Q5B_43A
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41A+Q5B_43B
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41B+Q5B_43B
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41B+Q5B_43B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41C+Q5B_43A
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41C+Q5B_43B
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A ne -1 and Q5B_43A ne -2 and Q5B_43A ne 995 and Q5B_43A ne -4))
compute HRACTT=Q5B_41D+Q5B_43A
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43B ne -1 and Q5B_43B ne -2 and Q5B_43B ne 995 and Q5B_43B ne -4))
compute HRACTT=Q5B_41D+Q5B_43B
else if ((Q5B_41A ne -4 and Q5B_41A ne -1 and Q5B_41A ne -2 and Q5B_41A ne 995) and (Q5B_43A eq -4 and Q5B_43B eq -4))
compute HRACTT=Q5B_41A
else if ((Q5B_41B ne -4 and Q5B_41B ne -1 and Q5B_41B ne -2 and Q5B_41B ne 995) and (Q5B_43A eq -4 and Q5B_43B eq -4))
compute HRACTT=Q5B_41B
else if ((Q5B_41C ne -4 and Q5B_41C ne -1 and Q5B_41C ne -2 and Q5B_41C ne 995) and (Q5B_43A eq -4 and Q5B_43B eq -4))
compute HRACTT=Q5B_41C
else if ((Q5B_41D ne -4 and Q5B_41D ne -1 and Q5B_41D ne -2 and Q5B_41D ne 995) and (Q5B_43A eq -4 and Q5B_43B eq -4))
compute HRACTT=Q5B_41D
else if ((Q5B_41A eq -1 or Q5B_41A eq -2 or Q5B_41B eq -1 or Q5B_41B eq -2 or Q5B_41C eq -1 or Q5B_41C eq -2 or Q5B_41D eq -1 or Q5B_41D eq -2) and (Q5B_43A eq -4 and Q5B_43B eq -4))
compute HRACTT=-2
else if ((Q5B_41A eq 995 or Q5B_41B eq 995 or Q5B_41C eq 995 or Q5B_41D eq 995) and (Q5B_43A eq -4 and Q5B_43B eq -4))
compute HRACTT=995
else if (Q5_40A eq -4)
compute HRACTT=-4
else if ((Q5_41A eq -1 or Q5_41B eq -1 or Q5_41C eq -1 or Q5_41D eq -1 or Q5_41A eq -2 or Q5_41B eq -2 or Q5_41C eq -2 or Q5_41D eq -2) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or Q5_43B eq -2))
compute HRACTT=-2
else if ((Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995) and (Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=995
else if ((Q5_41A eq -1 or Q5_41B eq -1 or Q5_41C eq -1 or Q5_41D eq -1 or Q5_41A eq -2 or Q5_41B eq -2 or Q5_41C eq -2 or Q5_41D eq -2) and (Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=995

```

else if ((Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995 or Q5_41A eq 995 or Q5_41B eq
995 or Q5_41C eq 995 or Q5_41D eq 995) and (Q5_43A eq -1 or Q5_43B eq -1 or Q5_43A eq -2 or
Q5_43B eq -2))
compute HRACTT=995
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and (Q5_43A eq -1 or Q5_43B eq
-1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=Q5_41A
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and (Q5_43A eq -1 or Q5_43B eq
-1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=Q5_41B
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and (Q5_43A eq -1 or Q5_43B eq
-1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=Q5_41C
else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and (Q5_43A eq -1 or Q5_43B eq
-1 or Q5_43A eq -2 or Q5_43B eq -2 or Q5_43A eq 995 or Q5_43B eq 995))
compute HRACTT=Q5_41D
else if ((Q5_43B ne -4 and Q5_43A ne -1 and Q5_43A ne -2 and Q5_43A ne 995) and (Q5_41A eq -1 or Q5_41A eq
-2 or Q5_41A eq 995 or Q5_41B eq -1 or Q5_41B eq -2 or Q5_41B eq 995 or Q5_41C eq -1 or Q5_41C eq -
2 or Q5_41C eq 995 or Q5_41D eq -1 or Q5_41D eq -2 or Q5_41D eq 995))
compute HRACTT=Q5_43A
else if ((Q5_43A ne -4 and Q5_43B ne -1 and Q5_43B ne -2 and Q5_43B ne 995) and (Q5_41A eq -1 or Q5_41A eq
-2 or Q5_41A eq 995 or Q5_41B eq -1 or Q5_41B eq -2 or Q5_41B eq 995 or Q5_41C eq -1 or Q5_41C eq -
2 or Q5_41C eq 995 or Q5_41D eq -1 or Q5_41D eq -2 or Q5_41D eq 995))
compute HRACTT=Q5_43B
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and (Q5_43A ne -1 and Q5_43A
ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT=Q5_41A+Q5_43A
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and (Q5_43B ne -1 and Q5_43B
ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT=Q5_41A+Q5_43B
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and (Q5_43A ne -1 and Q5_43A
ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT=Q5_41B+Q5_43A
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and (Q5_43B ne -1 and Q5_43B
ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT=Q5_41B+Q5_43B
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and (Q5_43A ne -1 and Q5_43A
ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT=Q5_41C+Q5_43A
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and (Q5_43B ne -1 and Q5_43B
ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT=Q5_41C+Q5_43B
else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and (Q5_43A ne -1 and Q5_43A
ne -2 and Q5_43A ne 995 and Q5_43A ne -4))
compute HRACTT=Q5_41D+Q5_43A
else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and (Q5_43B ne -1 and Q5_43B
ne -2 and Q5_43B ne 995 and Q5_43B ne -4))
compute HRACTT=Q5_41D+Q5_43B
else if ((Q5_41A ne -4 and Q5_41A ne -1 and Q5_41A ne -2 and Q5_41A ne 995) and Q5_43A eq -4 and Q5_43B
eq -4)
compute HRACTT=Q5_41A
else if ((Q5_41B ne -4 and Q5_41B ne -1 and Q5_41B ne -2 and Q5_41B ne 995) and Q5_43A eq -4 and Q5_43B
eq -4)
compute HRACTT=Q5_41B
else if ((Q5_41C ne -4 and Q5_41C ne -1 and Q5_41C ne -2 and Q5_41C ne 995) and Q5_43A eq -4 and Q5_43B
eq -4)
compute HRACTT=Q5_41C

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

else if ((Q5_41D ne -4 and Q5_41D ne -1 and Q5_41D ne -2 and Q5_41D ne 995) and Q5_43A eq -4 and Q5_43B
    eq -4)
compute HRACTT=Q5_41D
else if ((Q5_41A eq -1 or Q5_41A eq -2 or Q5_41B eq -1 or Q5_41B eq -2 or Q5_41C eq -1 or Q5_41C eq -2 or
    Q5_41D eq -1 or Q5_41D eq -2) and Q5_43A eq -4 and Q5_43B eq -4)
compute HRACTT=-2
else if ((Q5_41A eq 995 or Q5_41B eq 995 or Q5_41C eq 995 or Q5_41D eq 995) and Q5_43A eq -4 and Q5_43B
    eq -4)
compute HRACTT=995
else
    compute HRACTT=-3
end if

/* the following lines creates the var BUS2 */
do if (Q5_6 eq -4)
    compute BUS2=-4
else if (Q5_6 eq -1)
    compute BUS2=-1
else if (Q5_6 eq -2)
    compute BUS2=-2
else if (Q5_7 eq -1 or Q5_8 eq -1)
    compute BUS2=-1
else if (Q5_7 eq -2 or Q5_8 eq -2)
    compute BUS2=-2
else if (Q5_7 eq 1 or Q5_8 eq 1)
    compute BUS2=1
else if (Q5_8 eq 0)
    compute BUS2=2 /* this should normally be 0 */
else
    compute BUS2=-3
end if

/* the following lines create the var HRUSL1 */
do if (Q5B2_25 eq -4)
    compute HRUSL1=-4
else if (Q5B2_26A eq -1 or Q5B2_26C eq -1 or Q5B2_26A eq -2 or Q5B2_26C eq -2)
    compute HRUSL1=-2
else if (Q5B2_26A eq 995 or Q5B2_26C eq 995)
    compute HRUSL1=995
else if (Q5B2_26A eq 1)
    compute HRUSL1=Q5B2_26B
else if (Q5B2_26C eq 1)
    compute HRUSL1=Q5B2_26D
else if (Q5B_25 eq -4)
    compute HRUSL1=-4
else if (Q5B_26A eq -1 or Q5B_26C eq -1 or Q5B_26A eq -2 or Q5B_26C eq -2)
    compute HRUSL1=-2
else if (Q5B_26A eq 995 or Q5B_26C eq 995)
    compute HRUSL1=995
else if (Q5B_26A eq 1)
    compute HRUSL1=Q5B_26B
else if (Q5B_26C eq 1)
    compute HRUSL1=Q5B_26D
else if (Q5_25 eq -4)
    compute HRUSL1=-4
else if (Q5_26A eq -1 or Q5_26C eq -1 or Q5_26A eq -2 or Q5_26C eq -2)
    compute HRUSL1=-2
else if (Q5_26A eq 995 or Q5_26C eq 995)
    compute HRUSL1=995
else if (Q5_26A eq 1)
    compute HRUSL1=Q5_26B
else if (Q5_26C eq 1)
    compute HRUSL1=Q5_26D
else
    compute HRUSL1=-3
end if

/* the following lines create the var HRUSL2 */
do if (Q5B2_28A eq -4)
    compute HRUSL2=-4
else if (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B eq -2 or Q5B2_28D eq -2)
    compute HRUSL2=-2
else if (Q5B2_28B eq 995 or Q5B2_28D eq 995)
    compute HRUSL2=995
else if (Q5B2_28B eq 1)
    compute HRUSL2=Q5B2_28C
else if (Q5B2_28D eq 1)
    compute HRUSL2=Q5B2_28E
else if (Q5B_28A eq -4)
    compute HRUSL2=-4
else if (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2)
    compute HRUSL2=-2
else if (Q5B_28B eq 995 or Q5B_28D eq 995)
    compute HRUSL2=995
else if (Q5B_28B eq 1)
    compute HRUSL2=Q5B_28C
else if (Q5B_28D eq 1)
    compute HRUSL2=Q5B_28E
else if (Q5_28A eq -4)
    compute HRUSL2=-4
else if (Q5_28B eq -1 or Q5_28D eq -1 or Q5_28B eq -2 or Q5_28D eq -2)
    compute HRUSL2=-2
else if (Q5_28B eq 995 or Q5_28D eq 995)
    compute HRUSL2=995
else if (Q5_28B eq 1)
    compute HRUSL2=Q5_28C
else if (Q5_28D eq 1)
    compute HRUSL2=Q5_28E
else
    compute HRUSL2=-3
end if

```

```

/* the following lines create the var HRUSLT */
do if (Q5B2_25 eq -4)
compute HRUSLT=-4
else if ((Q5B2_26A eq -1 or Q5B2_26C eq -1 or Q5B2_26A eq -2 or Q5B2_26C eq -2) and (Q5B2_28B eq -1 or
      Q5B2_28D eq -1 or Q5B2_28B eq -2 or Q5B2_28D eq -2))
compute HRUSLT=-2
else if ((Q5B2_26A eq 995 or Q5B2_26C eq 995) and (Q5B2_28B eq 995 or Q5B2_28D eq 995))
compute HRUSLT=995
else if ((Q5B2_26A eq -1 or Q5B2_26C eq -1 or Q5B2_26A eq -2 or Q5B2_26C eq -2) and (Q5B2_28B eq 995 or
      Q5B2_28D eq 995))
compute HRUSLT=995
else if ((Q5B2_26A eq 995 or Q5B2_26C eq 995) and (Q5B2_28B eq -1 or Q5B2_28D eq -1 or Q5B2_28B eq -2 or
      Q5B2_28D eq -2))
compute HRUSLT=995
else if ((Q5B2_26A eq 995 or Q5B2_26C eq 995) and Q5B2_27 ne 1)
compute HRUSLT=995
else if ((Q5B2_26A eq -1 or Q5B2_26C eq -1 or Q5B2_26A eq -2 or Q5B2_26C eq -2) and Q5B2_27 ne 1)
compute HRUSLT=-2
else if (Q5B2_26A eq 1 and (Q5B2_28B eq -1 or Q5B2_28B eq -2 or Q5B2_28B eq 995 or Q5B2_28D eq -1 or
      Q5B2_28D eq -2 or Q5B2_28D eq 995))
compute HRUSLT=Q5B2_26B
else if (Q5B2_26C eq 1 and (Q5B2_28B eq -1 or Q5B2_28B eq -2 or Q5B2_28B eq 995 or Q5B2_28D eq -1 or
      Q5B2_28D eq -2 or Q5B2_28D eq 995))
compute HRUSLT=Q5B2_26D
else if (Q5B2_28B eq 1 and (Q5B2_26A eq -1 or Q5B2_26A eq -2 or Q5B2_26A eq 995 or Q5B2_26C eq -1 or
      Q5B2_26C eq -2 or Q5B2_26C eq 995))
compute HRUSLT=Q5B2_28C
else if (Q5B2_28D eq 1 and (Q5B2_26A eq -1 or Q5B2_26A eq -2 or Q5B2_26A eq 995 or Q5B2_26C eq -1 or
      Q5B2_26C eq -2 or Q5B2_26C eq 995))
compute HRUSLT=Q5B2_28E
else if (Q5B2_26A eq 1 and Q5B2_27 ne 1)           compute HRUSLT=Q5B2_26B
else if (Q5B2_26C eq 1 and Q5B2_27 ne 1)           compute HRUSLT=Q5B2_26D
else if (Q5B2_26A eq 1 and Q5B2_28B eq 1)          compute HRUSLT=Q5B2_26B+Q5B2_28C
else if (Q5B2_26A eq 1 and Q5B2_28D eq 1)          compute HRUSLT=Q5B2_26B+Q5B2_28E
else if (Q5B2_26C eq 1 and Q5B2_28B eq 1)          compute HRUSLT=Q5B2_26D+Q5B2_28C
else if (Q5B2_26C eq 1 and Q5B2_28D eq 1)          compute HRUSLT=Q5B2_26C+Q5B2_28E
else if (Q5B2_25 eq -4)                           compute HRUSLT=-4
else if ((Q5B_26A eq -1 or Q5B_26C eq -1 or Q5B_26A eq -2 or Q5B_26C eq -2) and (Q5B_28B eq -1 or Q5B_28D
      eq -1 or Q5B_28B eq -2 or Q5B_28D eq -2))
compute HRUSLT=-2
else if ((Q5B_26A eq 995 or Q5B_26C eq 995) and (Q5B_28B eq 995 or Q5B_28D eq 995))
compute HRUSLT=995
else if ((Q5B_26A eq -1 or Q5B_26C eq -1 or Q5B_26A eq -2 or Q5B_26C eq -2) and (Q5B_28B eq 995 or
      Q5B_28D eq 995))
compute HRUSLT=995
else if ((Q5B_26A eq 995 or Q5B_26C eq 995) and (Q5B_28B eq -1 or Q5B_28D eq -1 or Q5B_28B eq -2 or
      Q5B_28D eq -2))
compute HRUSLT=995
else if ((Q5B_26A eq 995 or Q5B_26C eq 995) and Q5B_27 ne 1)
compute HRUSLT=995
else if ((Q5B_26A eq -1 or Q5B_26C eq -1 or Q5B_26A eq -2 or Q5B_26C eq -2) and Q5B_27 ne 1)
compute HRUSLT=-2
else if (Q5B_26A eq 1 and (Q5B_28B eq -1 or Q5B_28B eq -2 or Q5B_28B eq 995 or Q5B_28D eq -1 or Q5B_28D
      eq -2 or Q5B_28D eq 995))
compute HRUSLT=Q5B_26B
else if (Q5B_26C eq 1 and (Q5B_28B eq -1 or Q5B_28B eq -2 or Q5B_28B eq 995 or Q5B_28D eq -1 or Q5B_28D
      eq -2 or Q5B_28D eq 995))

```

```

compute HRUSLT=Q5B_26D
else if (Q5B_28B eq 1 and (Q5B_26A eq -1 or Q5B_26A eq -2 or Q5B_26A eq 995 or Q5B_26C eq -1 or Q5B_26C
eq -2 or Q5B_26C eq 995))
compute HRUSLT=Q5B_28C
else if (Q5B_28D eq 1 and (Q5B_26A eq -1 or Q5B_26A eq -2 or Q5B_26A eq 995 or Q5B_26C eq -1 or Q5B_26C
eq -2 or Q5B_26C eq 995))
compute HRUSLT=Q5B_28E
else if (Q5B_26A eq 1 and Q5B_27 ne 1)           compute HRUSLT=Q5B_26B
else if (Q5B_26C eq 1 and Q5B_27 ne 1)           compute HRUSLT=Q5B_26D
else if (Q5B_26A eq 1 and Q5B_28B eq 1)          compute HRUSLT=Q5B_26B+Q5B_28C
else if (Q5B_26A eq 1 and Q5B_28D eq 1)          compute HRUSLT=Q5B_26B+Q5B_28E
else if (Q5B_26C eq 1 and Q5B_28B eq 1)          compute HRUSLT=Q5B_26D+Q5B_28C
else if (Q5B_26C eq 1 and Q5B_28D eq 1)          compute HRUSLT=Q5B_26C+Q5B_28E
else if (Q5_25 eq -4)                            compute HRUSLT=-4
else if ((Q5_26A eq -1 or Q5_26C eq -1 or Q5_26A eq -2 or Q5_26C eq -2) and (Q5_28B eq -1 or Q5_28D eq -1 or
Q5_28B eq -2 or Q5_28D eq -2))                  compute HRUSLT=-2
else if ((Q5_26A eq 995 or Q5_26C eq 995) and (Q5_28B eq 995 or Q5_28D eq 995))                  compute HRUSLT=995
else if ((Q5_26A eq -1 or Q5_26C eq -1 or Q5_26A eq -2 or Q5_26C eq -2) and (Q5_28B eq 995 or Q5_28D eq
995))                                         compute HRUSLT=995
else if ((Q5_26A eq 995 or Q5_26C eq 995) and (Q5_28B eq -1 or Q5_28D eq -1 or Q5_28B eq -2 or Q5_28D eq -
2))                                         compute HRUSLT=995
else if ((Q5_26A eq 995 or Q5_26C eq 995) and Q5_27 ne 1)                                     compute HRUSLT=995
else if ((Q5_26A eq -1 or Q5_26C eq -1 or Q5_26A eq -2 or Q5_26C eq -2) and Q5_27 ne 1)          compute HRUSLT=-2
else if (Q5_26A eq 1 and (Q5_28B eq -1 or Q5_28B eq -2 or Q5_28B eq 995 or Q5_28D eq -1 or Q5_28D eq -2 or
Q5_28D eq 995))                                compute HRUSLT=Q5_26B
else if (Q5_26C eq 1 and (Q5_28B eq -1 or Q5_28B eq -2 or Q5_28B eq 995 or Q5_28D eq -1 or Q5_28D eq -2 or
Q5_28D eq 995))                                compute HRUSLT=Q5_26D
else if (Q5_28B eq 1 and (Q5_26A eq -1 or Q5_26A eq -2 or Q5_26A eq 995 or Q5_26C eq -1 or Q5_26C eq -2 or
Q5_26C eq 995))                                compute HRUSLT=Q5_26C
else if (Q5_28D eq 1 and (Q5_26A eq -1 or Q5_26A eq -2 or Q5_26A eq 995 or Q5_26C eq -1 or Q5_26C eq -2 or
Q5_26C eq 995))                                compute HRUSLT=Q5_28C
compute HRUSLT=Q5_28E
else if (Q5_26A eq 1 and Q5_27 ne 1)             compute HRUSLT=Q5_26B
else if (Q5_26C eq 1 and Q5_27 ne 1)             compute HRUSLT=Q5_26D
else if (Q5_26A eq 1 and Q5_28B eq 1)            compute HRUSLT=Q5_26B+Q5_28C
else if (Q5_26A eq 1 and Q5_28D eq 1)            compute HRUSLT=Q5_26B+Q5_28E
else if (Q5_26C eq 1 and Q5_28B eq 1)            compute HRUSLT=Q5_26D+Q5_28C
else if (Q5_26C eq 1 and Q5_28D eq 1)            compute HRUSLT=Q5_26C+Q5_28E
else if (Q5_26C eq 1 and Q5_28D eq 1)            compute HRUSLT=-3
else                                         end if

/* the following lines create the var HRFTPT */
do if (Q5B2_30C eq -4)                          compute HRFTPT=-4
else if (Q5B2_30F eq -1 or Q5B2_30G eq -1 or Q5B2_30H eq -1 or Q5B2_30I eq -1)                  compute HRFTPT=-1
else if (Q5B2_30F eq -2 or Q5B2_30G eq -2 or Q5B2_30H eq -2 or Q5B2_30I eq -2)                  compute HRFTPT=-2
else if (Q5B2_30F ne -4)                        compute HRFTPT=Q5B2_30F
else if (Q5B2_30G ne -4)                        compute HRFTPT=Q5B2_30G
else if (Q5B2_30H ne -4)                        compute HRFTPT=Q5B2_30H
else if (Q5B2_30I ne -4)                        compute HRFTPT=Q5B2_30I

```

```

else if (Q5B_30C eq -4)           compute HRFTPT=-4
else if (Q5B_30F eq -1 or Q5B_30G eq -1 or Q5B_30H eq -1 or Q5B_30I eq -1)   compute HRFTPT=-1
else if (Q5B_30F eq -2 or Q5B_30G eq -2 or Q5B_30H eq -2 or Q5B_30I eq -2)   compute HRFTPT=-2
else if (Q5B_30F ne -4)          compute HRFTPT=Q5B_30F
else if (Q5B_30G ne -4)          compute HRFTPT=Q5B_30G
else if (Q5B_30H ne -4)          compute HRFTPT=Q5B_30H
else if (Q5B_30I ne -4)          compute HRFTPT=Q5B_30I
else if (Q5_30C eq -4)           compute HRFTPT=-4
else if (Q5_30F eq -1 or Q5_30G eq -1 or Q5_30H eq -1 or Q5_30I eq -1)       compute HRFTPT=-1
else if (Q5_30F eq -2 or Q5_30G eq -2 or Q5_30H eq -2 or Q5_30I eq -2)       compute HRFTPT=-2
else if (Q5_30F ne -4)          compute HRFTPT=Q5_30F
else if (Q5_30G ne -4)          compute HRFTPT=Q5_30G
else if (Q5_30H ne -4)          compute HRFTPT=Q5_30H
else if (Q5_30I ne -4)          compute HRFTPT=Q5_30I
else                            compute HRFTPT=-3
                                end if

/* the following lines create the var HRWANT */
do if (Q5B2_33 ne -4)            compute HRWANT=Q5B2_33
else if (Q5B_33 ne -4)           compute HRWANT=Q5B_33
else                            compute HRWANT=Q5_33
                                end if

/* the following lines create the var HRRSN1 */
do if (Q5B2_34 ne -4)            compute HRRSN1=Q5B2_34
else if (Q5B_34 ne -4)           compute HRRSN1=Q5B_34
else                            compute HRRSN1=Q5_34
                                end if

/* the following lines create the var HRCK7 */
compute HRCK7=-4
do if ((BUS2 eq 2 or BUS2 eq -2 or BUS2 eq -1) and (HRACT1 LT 15 or HRACT1 eq -2))  compute HRCK7=1
compute HRCK7=1
else if ((BUS2 eq 2 or BUS2 eq -2 or BUS2 eq -1) and HRACT1 GE 15)                  compute HRCK7=2
compute HRCK7=2
else if ((HRUSLT GE 35 or HRFTPT eq 1) and HRACTT LT 35 and (HRACT1 ne -1 or HRACT1 ne -2 or
      HRACT2 ne -1 or HRACT2 ne -2))        compute HRCK7=3
compute HRCK7=3
else if (HRWANT eq 1 and HRACTT LT 35 and (HRRSN1 eq 1 or HRRSN1 eq 2 or HRRSN1 eq 3))  compute HRCK7=4
compute HRCK7=4
else                            compute HRCK7=5
                                end if

/* the following lines create the var ABSOT */
do if (HRACT1 eq 0 and HRACT2 eq 0)  compute ABSOT=1
else                            compute ABSOT=0
                                end if

/* the following lines create the var HRCK6 */
compute HRCK6=-4
do if ((HRACT1 eq 0 and HRACT2 eq 0) and (BUS2 eq 2 or BUS2 eq -1 or BUS2 eq -2))  compute HRCK6=1
compute HRCK6=1
else if (ABSOT eq 1)                compute HRCK6=2
else                            compute HRCK6=3
                                end if

/* the following lines create the var WK */
compute WK=-4
do if (Q5_2 ne -4)                 compute WK=Q5_2
else if (Q5_3 ne -4)               compute WK=Q5_3
                                end if

/* the following lines create the var RETOT */
do if (WK eq 2)                   compute RETOT=1

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```

else                                         compute RETOT=0                         end if

/* the following lines rename the vars */
compute BUS1=Q5_5                           compute RET1=Q5_10
compute IO1COW=Q5_JUMP                      compute NLFACt=Q5_93

/* the following lines create additional variables */
do if (Q5B2_22 ne -4)                      compute ABSPD=Q5B2_22
else if (Q5B_22 ne -4)                     compute ABSPD=Q5B_22
else                                         compute ABSPD=Q5_22                         end if

do if (Q5B2_65 ne -4)                      compute LKAVL=Q5B2_65
else if (Q5B_65 ne -4)                     compute LKAVL=Q5B_65
else                                         compute LKAVL=Q5_65                         end if

do if (Q5B2_51 ne -4)                      compute LAYAVL=Q5B2_51
else if (Q5B_51 ne -4)                     compute LAYAVL=Q5B_51
else                                         compute LAYAVL=Q5_51                         end if

do if (Q5B2_66 ne -4)                      compute LKAVR=Q5B2_66
else if (Q5B_66 ne -4)                     compute LKAVR=Q5B_66
else                                         compute LKAVR=Q5_66                         end if

do if (Q5B2_52 ne -4)                      compute LAYAVR=Q5B2_52
else if (Q5B_52 ne -4)                     compute LAYAVR=Q5B_52
else                                         compute LAYAVR=Q5_52                         end if

do if (Q5B2_60 ne -4)                      compute LK=Q5B2_60
else if (Q5B_60 ne -4)                     compute LK=Q5B_60
else                                         compute LK=Q5_60                           end if

do if (Q5B2_11 ne -4)                      compute DIS1=Q5B2_11
else if (Q5B_11 ne -4)                     compute DIS1=Q5B_11
else                                         compute DIS1=Q5_11                         end if

do if (Q5B2_12 ne -4)                      compute DIS2=Q5B2_12
else if (Q5B_12 ne -4)                     compute DIS2=Q5B_12
else                                         compute DIS2=Q5_12                           end if

/* the following lines create the var DWWNTO */
do if (RET1 eq 1)                           compute DWWNTO=1
else if (Q5B2_78 eq 1)                     compute DWWNTO=1
else if (Q5B2_78 eq 0)                     compute DWWNTO=2
else if (Q5B2_78 eq -2)                    compute DWWNTO=-2
else if (Q5B2_78 eq -1)                    compute DWWNTO=-1
else if (Q5B2_78 eq 3)                     compute DWWNTO=3
else if (Q5B2_78 eq 4)                     compute DWWNTO=4
else if (Q5B_78 eq 1)                      compute DWWNTO=1
else if (Q5B_78 eq 0)                      compute DWWNTO=2
else if (Q5B_78 eq -2)                     compute DWWNTO=-2
else if (Q5B_78 eq -1)                     compute DWWNTO=-1
else if (Q5B_78 eq 3)                      compute DWWNTO=3
else if (Q5B_78 eq 4)                      compute DWWNTO=4
else if (Q5_78 eq 1)                       compute DWWNTO=1
else if (Q5_78 eq 0)                       compute DWWNTO=2
else if (Q5_78 eq -2)                      compute DWWNTO=-2
else if (Q5_78 eq -1)                      compute DWWNTO=-1

```

Appendix 1: Employment Status Recode (ESR) Variable Creation

```
else if (Q5_78 eq 3)           compute DWWNTO=3
else if (Q5_78 eq 4)           compute DWWNTO=4
else                           compute DWWNTO=-3
                                end if

/* the following lines create the var DIS */
do if (Q5_9A4 eq -4)          compute DIS=-4
else if (Q5_9D eq -1 or Q5_9E eq -1 or Q5_9F eq -1 or Q5_9G eq -1) or (Q5_9D eq -2 or Q5_9E eq -2 or Q5_9F eq
-2 or Q5_9G eq -2)
compute DIS=-2
else if (Q5_9A4 eq 1 and Q5_9D eq 1)   compute DIS=1
else if (Q5_9D eq 0)             compute DIS=0
else if (Q5_9A5 eq 1 and Q5_9E eq 1)   compute DIS=1
else if (Q5_9E eq 0)             compute DIS=0
else if (Q5_9A6 eq 1 and Q5_9F eq 1)   compute DIS=1
else if (Q5_9F eq 0)             compute DIS=0
else if (Q5_9A7 eq 1 and Q5_9G eq 1)   compute DIS=1
else if (Q5_9G eq 0)             compute DIS=0
else                             compute DIS=-3
                                end if

/* the following lines create IO1COW */
do if (Q5_JUMP eq 1)           compute IO1COW=8
else                           compute IO1COW=0
                                end if

/* the following lines create the ESR var approximating MLR */
do if ((WK eq 1 and (HRCK6 eq 3 or HRCK6 eq -4 or HRCK6 eq 99 or HRCK6 eq 999)) or (BUS1 eq 1 and
(HRCK7 eq 2 or HRCK7 eq 3 or HRCK7 eq 4 or HRCK7 eq 5 or HRCK7 eq -4)))
compute ESR=1
else if (ABSPD eq 1 or ABSPD eq 0 or ABSPD eq -1 or ABSPD eq -2)
compute ESR=2
else if ((LAYAVL eq 1 or LAYAVL eq -1 or LAYAVL eq -2) or LAYAVR eq 1)
compute ESR=3
else if ((LKAVL eq 1 or LKAVL eq -1 or LKAVL eq -2) or (LKAVR eq 1 or LKAVR eq 2))
compute ESR=4
else if ((WK eq 3 and AGE GE 50 and RET1 eq 1 and LK eq 1 and LKAVL eq 0 and (LKAVR eq 3 or LKAVR eq 4 or
LKAVR eq -1 or LKAVR eq -2)) or (RET1 eq 0 or LK eq 3 or DWWNTO eq 3) or NLFACt eq 5 or RETOT
eq 1)
compute ESR=5
else if (DIS1 eq 1 or DIS2 eq 1 or DIS eq 1) compute ESR=6
else if (Q5_JUMP eq 1)           compute ESR=8
/* else if (IO1COW eq 8) */      /* compute ESR=7 */
else                           compute ESR=7
                                end if

/* COLLAPSED EMPLOYMENT STATUS RECODE 1998 */

do if (ESR = -5)                compute ESRC=-5
else if (ESR = 1 or ESR = 2)     compute ESRC=1
else if (ESR = 3 or ESR = 4)     compute ESRC=2
else if (ESR >= 4 and ESR <= 7) compute ESRC=3
else                           compute ESRC=ESR
                                end if

if (Q5_JUMP = 1) ESRC=4
```

NLSY79 APPENDIX 2:
TOTAL NET FAMILY INCOME VARIABLE
CREATION: 1979-2000

VARIABLE CREATION: TOTAL NET FAMILY INCOME 1979–2000

```

/* 1979-83 /
DCL 1 FAMILY_INCOME (5),
      5 CPS,
      5 FAMINC,
      5 INC,
      5 MILS,
      5 PUBLIC,
      5 S,
      5 TABLE (3,2,15),
      5 WELF,
      5 AFDC,
      5 ED,
      5 FARM,
      5 INCOME,
      5 MN (3),
      5 R,
      5 SEI,
      5 UI,
      5 WPS,
      5 ALIM,
      5 EDSS,
      5 FOOD,
      5 LEVEL,
      5 OTHER,
      5 RELREG,
      5 SEIS,
      5 UIS,
      5 WPSS,
      5 CHSP,
      5 FAMILY,
      5 GIFT,
      5 MIL,
      5 POVERTY,
      5 RELWEL,
      5 SUMN,
      5 VET,
      5 YOUTH;

/* 1979 VARIABLES */
INC(1)=R(1903.10);
if R(1547.)=-4 then MIL(1)=0; else MIL=R(1547.);
if R(1554.)=-4 then WPS(1)=0; else WPS(1)=R(1554.);
if R(1560.)=-4 then SEI(1)=0; else SEI(1)=R(1560.);
if R(1588.)=-4 & R(1590.)=-4 then UI(1)=0;
else if R(1588.)>0 & R(1590.)>0 then UI(1)=R(1588.) * R(1590.);
else UI(1)=R(1590.);
MILS(1)=0;
if R(1555.)=-4 then WPSS(1)=0; else WPSS(1)=R(1555.);
if R(1561.)=-4 then SEIS(1)=0; else SEIS(1)=R(1561.);
if R(1589.)=-4 & R(1591.)=-4 then UIS(1)=0;
else if R(1589.)>0 & R(1591.)>0 then UIS(1)=R(1589.) * R(1591.);
else UIS(1)=R(1591.);
if R(1594.)=-4 then ALIM(1)=0; else ALIM(1)=R(1594.);
CHSP(1)=0;
if R(1608.)=-4 then AFDC(1)=0;
else do; B=0;
do J=1596. to 1607.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(1608.)<0 then AFDC(1)=R(1608.);
else if B<=0 then AFDC(1)=-3; else AFDC(1)=R(1608.) * B;
end;
if R(1623.)=-4 then FOOD(1)=0;
else do; B=0;
do J=1610. to 1621.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if B<=0 then FOOD(1)=-3;
else if R(1622.)>=0 & R(1623.)>0 then FOOD(1)=(R(1623.)-R(1622.)) * B; else FOOD(1)=-3;
end;
WELF(1)=0;
if R(1640.)=-4 then PUBLIC(1)=0;
else do; B=0;
do J=1628. to 1639.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(1640.)<0 then PUBLIC(1)=R(1640.);
else if B<=0 then PUBLIC(1)=-3; else PUBLIC(1)=R(1640.) * B;
end;
if R(1645.)=-4 then ED(1)=0; else ED(1)=R(1645.);
if R(1646.)=-4 then EDSS(1)=0; else EDSS(1)=R(1646.);
VET(1)=0;
if R(1649.)=-4 then GIFT(1)=0; else GIFT(1)=R(1649.);
if R(1651.)=-4 then OTHER(1)=0; else OTHER(1)=R(1651.);
RELWEL(1)=0;
if R(1664.)=-4 then RELREG(1)=0; else RELREG(1)=R(1664.);
```

Appendix 2: Total Net Family Income Variable Creation

```

/*1979 FAMILY POVERTY STATUS, 1979 FAMILY POVERTY LEVEL, & 1979 FAMILY POVERTY FLAG*/
DCL PUBID PIC '99999';
    DCL S79      fixed DEC(9);      FAMILY79   fixed DEC(9);      FARM79     fixed DEC(9);
    SCREENSZ    fixed DEC(9);      FLAG        fixed DEC(9);      INCOME79   float DEC(6);
    CAL_INCOME  float DEC(6);     LEVEL79    float DEC(6);      STATE79    float DEC(6);
    CPS79       float DEC(6);     R79        float DEC(6);      TABLE(3,2,15) float DEC(6);
    R(1.10)     float DEC(6);     R(2179.)  float DEC(6);      R(1691.)   float DEC(6);
    R(1697.)   float DEC(6);     R(1743.)  float DEC(6);      R(1916.10) float DEC(6);
    R(1921.)   float DEC(6);     R(1919.)  float DEC(6);      FAMSZ79   float DEC(6);
    WPS         float DEC(6);     WPSS       float DEC(6);      SEI        float DEC(6);
    SEIS        float DEC(6);     MIL        float DEC(6);     OTHER      float DEC(6);
    RELINC     float DEC(6);     GIFT       float DEC(6);

LEVEL79=-4;
do I=1 to 3;
  do J=1 to 2;
    do K=1 to 15; TABLE(I,J,K)=-4; end;
  end;
end;
TABLE(1,1,1)=3400;           TABLE(2,1,1)=4270;           TABLE(3,1,1)=3930;
TABLE(1,2,1)=2910;           TABLE(2,2,1)=3650;           TABLE(3,2,1)=3350;
do J=2 to 15;
  TABLE(1,1,J)=TABLE(1,1,J-1)+1100;          TABLE(2,1,J)=TABLE(2,1,J-1)+1370;
  TABLE(3,1,J)=TABLE(3,1,J-1)+1260;          TABLE(1,2,J)=TABLE(1,2,J-1)+930;
  TABLE(2,2,J)=TABLE(2,2,J-1)+1160;          TABLE(3,2,J)=TABLE(3,2,J-1)+1070;
end;

FARM79=1;
if R(1919.)>4 then FARM79=2;
if R(1743.)>0 then STATE79=MOD(R(1743.),100); else STATE79=R(1743.);
S79=1;
if STATE79=2 then S79=2; else if STATE79=15 then S79=3;

/*To create family size (FAMSZ79) search thru the household enumeration increment family size, if the relationship to
the youth is a relative. Do not increase family size if the code is <0 or (ge 33 & le 36) or =45 or =46 or *ge 50 & le 54)*/
if FAMSZ79=0 then FAMILY79=1;
else FAMILY79=FAMSZ79;

INCOME79=R(2179.);
R79=0;
if R(1691.)>=0 then R79=R79+R(1691.);
if R(1697.)>=0 then R79=R79+R(1697.);

if FAMILY79>0 then do;
  LEVEL79=TABLE(S79,FARM79,FAMILY79);
  if INCOME79>LEVEL79 then CPS79=0;
  else if INCOME79>= 0 & INCOME79<=LEVEL79 then CPS79=1;
  else if R79>TABLE(S79,FARM79,FAMILY79) then CPS79=0;
  else CPS79=-3;
end;
else do;
  if INCOME79>TABLE(S79,FARM79,15) then CPS79=0;
  else if INCOME79>=0 & INCOME79<=TABLE(S79,FARM79,1) then CPS79=1;
  else CPS79=-3;
end;
FLAG=0;

```

```
CAL_INCOME=0;
if CPS79=-3 & INCOME79<0 then do;
  if WPS>=0 then CAL_INCOME=CAL_INCOME + WPS;
  if WPSS>=0 then CAL_INCOME=CAL_INCOME + WPSS;
  if SEI>=0 then CAL_INCOME=CAL_INCOME + SEI;
  if SEIS>=0 then CAL_INCOME=CAL_INCOME + SEIS;
  if MIL>=0 then CAL_INCOME=CAL_INCOME + MIL;
  if OTHER>=0 then CAL_INCOME=CAL_INCOME + OTHER;
  if RELINC>=0 then CAL_INCOME=CAL_INCOME + RELINC;
  if GIFT>=0 then CAL_INCOME=CAL_INCOME + GIFT;
  if CAL_INCOME>0 then do;
    if CAL_INCOME>LEVEL79 then do;
      CPS79=0;
      FLAG=1;
    end;
  end;
end;
if CPS79=-3 & R(1916.10)>0 & R(1921.)>0 then do;
  SCREENSZ=R(1921.);
  if SCREENSZ=-4 then FAMILY79=1;
  else FAMILY79=SCREENSZ;
  LEVEL79=TABLE(S79,FARM79,FAMILY79);
  if R(1916.10)>LEVEL79 then do;
    CPS79=0;
    FLAG=2;
  end;
  else if R(1916.10)<=LEVEL79 then do;
    CPS79=1;
    FLAG=2;
  end;
end;
CPS79=R(2179.10);
LEVEL79=R(2179.20);
FLAG79=R(2179.30);
```

Appendix 2: Total Net Family Income Variable Creation

```

/* 1980 VARIABLES */

if R(4052.)=-4 then INC(2)=-5;
if R(3120.)=-4 then MIL(2)=0;
if R(3123.)=-4 then WPS(2)=0;
if R(3126.)=-4 then SEI(2)=0;
if R(3146.)=-4 & R(3132.)=-4 then UI(2)=0;
else if R(3146.)>0 & R(3132.)>0 then UI(2)=R(3146.) * R(3132.);
else UI(2)=R(3132.);
if R(3122.)=-4 then MILS(2)=0;
if R(3127.10)=-4 then WPSS(2)=0;
if R(3130.)=-4 then SEIS(2)=0;
if R(3160.)=-4 & R(3161.)=-4 then UIS(2)=0;
else if R(3160.)>0 & R(3161.)>0 then UIS(2)=R(3160.) * R(3161.);
else UIS(2)=R(3161.);
if R(3164.)=-4 then ALIM(2)=0;
CHSP(2)=0;
if R(3178.)=-4 then AFDC(2)=0;
else do; B=0;
do J=3166. to 3177.; if R(J)>0 & B^=na then B=B+1; else if R(J)<-4 & R(J)<0 then B=na; end;
if R(3178.)<0 then AFDC(2)=R(3178.);
else if B<=0 then AFDC(2)=-3;
else AFDC(2)=R(3178.) * B;
end;
if R(3192.)=-4 then FOOD(2)=0;
else do; B=0;
do J=3180. to 3191.; if R(J)>0 & B^=na then B=B+1; else if R(J)<-4 & R(J)<0 then B=na; end;
if R(3192.)=-4 then FOOD(2)=R(3192.);
else if B<=0 then FOOD(2)=-3;
else FOOD(2)=R(3192.) * B;
end;
if R(3206.)=-4 then WELF(2)=0;
else do; B=0;
do J=3194. to 3205.; if R(J)>0 & B^=na then B=B+1; else if R(J)<-4 & R(J)<0 then B=na; end;
if R(3206.)<0 then WELF(2)=R(3206.);
else if B<=0 then WELF(2)=-3;
else WELF(2)=R(3206.) * B;
end;
if R(3220.)=-4 then PUBLIC(2)=0;
else do; B=0;
do J=3208. to 3219.; if R(J)>0 & B^=na then B=B+1; else if R(J)<-4 & R(J)<0 then B=na; end;
if R(3220.)<0 then PUBLIC(2)=R(3220.);
else if B<=0 then PUBLIC(2)=-3;
else PUBLIC(2)=R(3220.) * B;
end;
if R(3225.)=-4 then ED(2)=0;
if R(3226.)=-4 then EDSS(2)=0;
if R(3228.)=-4 then VET(2)=0;
if R(3233.)=-4 then GIFT(2)=0;
if R(3235.)=-4 then OTHER(2)=0;
if R(3244.)=-4 then RELWEL(2)=0;
if R(3251.10)=-4 then RELREG(2)=0;
if R(4047.)=-4 then POVERTY(2)=0;
FAMILY(2)=0;
do J=3939. to 4023. BY 6;
if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=53) then A=1;
else FAMILY(2)=FAMILY(2)+1;
end;

```

Appendix 2: Total Net Family Income Variable Creation

```

/* 1981 VARIABLES */

if R(6146.)<=0 then INC(3)=-5;                                else INC(3)=R(6138.10);
YOUTH(3)=0;          TABLE(3,1,1,1)=4320; TABLE(3,2,1,1)=5400; TABLE(3,3,1,1)=4970;
                     TABLE(3,1,2,1)=3690; TABLE(3,2,2,1)=4660; TABLE(3,3,2,1)=4270;
do J=2 to 15;        TABLE(3,1,1,J)=TABLE(3,1,1,J-1) + 1380; TABLE(3,2,1,J)=TABLE(3,2,1,J-1) + 1730;
                     TABLE(3,3,1,J)=TABLE(3,3,1,J-1) + 1590; TABLE(3,1,2,J)=TABLE(3,1,2,J-1) + 1170;
                     TABLE(3,2,2,J)=TABLE(3,2,2,J-1) + 1450; TABLE(3,3,2,J)=TABLE(3,3,2,J-1) + 1340;
end;
FARM(3)=1;           if R(6125.)>4 then FARM(3)=2; S(3)=1;
if R(6028.)>0 then do; if MOD(R(6028.),100)=2 then S(3)=2; if MOD(R(6028.),100)=15 then S(3)=3; end;
LEVEL(3)=-4; if R(4825.)=-4 then MIL(3)=0;                  else MIL(3)=R(4825.);
if R(4826.)=-4 then WPS(3)=0;                               else WPS(3)=R(4826.);
if R(4832.)=-4 then SEI(3)=0;                               else SEI(3)=R(4832.);
if R(4849.)=-4 & R(4850.)=-4 then UI(3)=0;                else if R(4849.)>0 & R(4850.)>0 then UI(3)=R(4849.) * R(4850.); else UI(3)=R(4850.);
if R(4828.)=-4 then MILS(3)=0;                            else MILS(3)=R(4828.);
if R(4829.10)=-4 then WPSS(3)=0;                          else WPSS(3)=R(4829.10);
if R(4835.)=-4 then SEIS(3)=0;                            else SEIS(3)=R(4835.);
if R(4865.)=-4 & R(4866.)=-4 then UIS(3)=0;              else if R(4865.)>0 & R(4866.)>0 then UIS(3)=R(4865.) * R(4866.); else UIS(3)=R(4866.);
if R(4869.)=-4 then ALIM(3)=0;                           else ALIM(3)=R(4869.);
CHSP(3)=0;                                                 if R(4883.)=-4 then AFDC(3)=0;
else do; B=0;                                            else do; B=0;
do J=4871. to 4882.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(4883.)<0 then AFDC(3)=R(4883.);                      else if B<=0 then AFDC(3)=-3; else AFDC(3)=R(4883.) * B;
end;
if R(4897.)=-4 then FOOD(3)=0;                           else if R(4897.)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(4897.)<0 then FOOD(3)=R(4897.);                      else if B<=0 then FOOD(3)=-3; else FOOD(3)=R(4897.) * B;
end;
if R(4911.)=-4 then WELF(3)=0;                           else if B<=0 then WELF(3)=-3; else WELF(3)=R(4911.) * B;
end;
if R(4925.)=-4 then PUBLIC(3)=0;                         else do; B=0;
do J=4913. to 4924.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(4925.)<0 then PUBLIC(3)=R(4925.);                      else if B<=0 then PUBLIC(3)=-3; else PUBLIC(3)=R(4925.) * B;
end;
if R(4930.)=-4 then ED(3)=0;                            else ED(3)=R(4930.);
if R(4931.)=-4 then EDSS(3)=0;                          else EDSS(3)=R(4931.);
if R(4933.)=-4 then VET(3)=0;                           else VET(3)=R(4933.);
if R(4938.)=-4 then GIFT(3)=0;                          else GIFT(3)=R(4938.);
if R(4940.)=-4 then OTHER(3)=0;                         else OTHER(3)=R(4940.);
if R(4949.)=-4 then RELWEL(3)=0;                        else RELWEL(3)=R(4949.);
if R(4956.10)=-4 then RELREG(3)=0;                      else RELREG(3)=R(4956.10);
if R(6140.)=-4 then POVERTY(3)=0;                       else POVERTY(3)=R(6140.);
FAMILY(3)=0;                                              do J=6033. to 6117. BY 6;
if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=53) then A=1;
else FAMILY(3)=FAMILY(3)+1;                             end;

```

Appendix 2: Total Net Family Income Variable Creation

```

/* 1982 VARIABLES */

if R(8967.)<=0 then INC(4)=-5; else INC(4)=R(8304.);
YOUTH(4)=0; if R(7986.)>=0 then YOUTH(4)=R(7986.);
if R(7988.)>=0 then YOUTH(4)=Y(4)+R(7988.);
if R(7992.)>=0 then YOUTH(4)=Y(4)+R(7992.);
TABLE(4,1,1,1)=4760; TABLE(4,2,1,1)=5990; TABLE(4,3,1,1)=5500;
TABLE(4,1,2,1)=4080; TABLE(4,2,2,1)=5110; TABLE(4,3,2,1)=4700;
do J=2 to 15; TABLE(4,1,1,J)=TABLE(4,1,1,J-1)+1530; TABLE(4,2,1,J)=TABLE(4,2,1,J-1)+1900;
TABLE(4,3,1,J)=TABLE(4,3,1,J-1)+1750; TABLE(4,1,2,J)=TABLE(4,1,2,J-1)+1290;
TABLE(4,2,2,J)=TABLE(4,2,2,J-1)+1610; TABLE(4,3,2,J)=TABLE(4,3,2,J-1)+1480;
end;
FARM(4)=1; if R(8291.)>4 then FARM(4)=2; S(4)=1;
if R(8169.)=2 then S(4)=2; else if R(8169.)=15 then S(4)=3;
LEVEL(4)=-4; if R(7820.)=-4 then MIL(4)=0; else MIL(4)=R(7820.);
if R(7821.)=-4 then WPS(4)=0; else WPS(4)=R(7821.);
if R(7824.)=-4 then SEI(4)=0; else SEI(4)=R(7824.);
if R(7838.)=-4 & R(7839.)=-4 then UI(4)=0;
else if R(7838.)>0 & R(7839.)>0 then UI(4)=R(7838.) * R(7839.); else UI(4)=R(7839.);
if R(7842.)=-4 then MILS(4)=0; else MILS(4)=R(7842.);
if R(7843.)=-4 then WPSS(4)=0; else WPSS(4)=R(7843.);
if R(7846.)=-4 then SEIS(4)=0; else SEIS(4)=R(7846.);
if R(7860.)=-4 & R(7861.)=-4 then UIS(4)=0;
else if R(7860.)>0 & R(7861.)>0 then UIS(4)=R(760.) * R(7861.); else UIS(4)=R(7861.);
if R(7864.)=-4 then ALIM(4)=0; else ALIM(4)=R(7864.);
if R(7869.)=-4 then CHSP(4)=0; else CHSP(4)=R(7869.);
if R(7885.)=-4 then AFDC(4)=0; else do; B=0;
do J=7873. to 7884.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(7885.)<0 then AFDC(4)=R(7885.);
else if B<=0 then AFDC(4)=-3; else AFDC(4)=R(7885.) * B;
end;
if R(7899.)=-4 then FOOD(4)=0;
else do; B=0; do J=7887. to 7898.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(7899.)<0 then FOOD(4)=R(7899.);
else if B<=0 then FOOD(4)=-3; else FOOD(4)=R(7899.) * B;
end;
if R(7913.)=-4 then WELF(4)=0;
else do; B=0; do J=7901. to 7912.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(7913.)<0 then WELF(4)=R(7913.);
else if B<=0 then WELF(4)=-3; else WELF(4)=R(7913.) * B;
end;
if R(7927.)=-4 then PUBLIC(4)=0;
else do; do J=7915. to 7926.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(7927.)<0 then PUBLIC(4)=R(7927.);
else if B<=0 then PUBLIC(4)=-3; else PUBLIC(4)=R(7927.) * B;
end;
if R(7932.)=-4 then ED(4)=0; else ED(4)=R(7932.);
if R(7933.)=-4 then EDSS(4)=0; else EDSS(4)=R(7933.);
if R(7935.)=-4 then VET(4)=0; else VET(4)=R(7935.);
if R(7939.)=-4 then GIFT(4)=0; else GIFT(4)=R(7939.);
if R(7941.)=-4 then OTHER(4)=0; else OTHER(4)=R(7941.);
if R(7949.)=-4 then RELWEL(4)=0; else RELWEL(4)=R(7949.);
if R(7956.)=-4 then RELREG(4)=0; else RELREG(4)=R(7956.);
if R(8306.)=-4 then POVERTY(4)=0; else POVERTY(4)=R(8306.);
FAMILY(4)=0; do J=8177. to 8275. BY 7;
if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=53) then A=1;
else FAMILY(4)=FAMILY(4)+1;
end;

```

/* 1983 VARIABLES */

```

if R(11444.)<=0 then INC(5)=-5;                                else INC(5)=R(10777.);
YOUTH(5)=0;          TABLE(5,1,1,1)=4910; TABLE(5,2,1,1)=6160; TABLE(5,3,1,1)=5670;
                     TABLE(5,1,2,1)=4200; TABLE(5,2,2,1)=5290; TABLE(5,3,2,1)=4850;
do J=2 to 15;        TABLE(5,1,1,J)=TABLE(5,1,1,J-1)+1620; TABLE(5,2,1,J)=TABLE(5,2,1,J-1)+2020;
                     TABLE(5,3,1,J)=TABLE(5,3,1,J-1)+1860; TABLE(5,1,2,J)=TABLE(5,1,2,J-1)+1370;
                     TABLE(5,2,2,J)=TABLE(5,2,2,J-1)+1700; TABLE(5,3,2,J)=TABLE(5,3,2,J-1)+1570;
end;
FARM(5)=1;           if R(10764.)>4 then FARM(5)=2;
S(5)=1; if R(10548.)=2 then S(5)=2;                           else if R(10548.)=15 then S(5)=3;
LEVEL(5)=-4; if R(10239.)=-4 then MIL(5)=0;                 else MIL(5)=R(10239.);
if R(10240.)=-4 then WPS(5)=0;                               else WPS(5)=R(10240.);
if R(10243.)=-4 then SEI(5)=0;                             else SEI(5)=R(10243.);
if R(10257.)=-4 & R(10258.)=-4 then UI(5)=0;               else if R(10257.)>0 & R(10258.)>0 then UI(5)=R(10257.) * R(10258.);
else if R(10258.)<0 then UI(5)=R(10258.);                  else UI(5)=-3;
if R(10261.)=-4 then MILS(5)=0;                            else MILS(5)=R(10261.);
if R(10262.)=-4 then WPSS(5)=0;                            else WPSS(5)=R(10262.);
if R(10265.)=-4 then SEIS(5)=0;                            else SEIS(5)=R(10265.);
if R(10279.)=-4 & R(10280.)=-4 then UIS(5)=0;             else if R(10279.)>0 & R(10280.)>0 then UIS(5)=R(10279.) * R(10280.);
else if R(10280.)<0 then UIS(5)=R(10280.);                  else UIS(5)=-3;
if R(10283.)=-4 then ALIM(5)=0;                            else ALIM(5)=R(10283.);
if R(10288.)=-4 then CHSP(5)=0;                            else CHSP(5)=R(10288.);
if R(10304.)=-4 then AFDC(5)=0;                           else do; B=0;
do J=10292. to 10303.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(10304.)<0 then AFDC(5)=R(10304.);                   else AFDC(5)=R(10304.) * B;
end;
if R(10318.)=-4 then FOOD(5)=0;                           else do; B=0;
do J=10306. to 10317.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(10318.)<0 then FOOD(5)=R(10318.);                   else FOOD(5)=R(10318.) * B;
end;
if R(10332.)=-4 then WELF(5)=0;                           else do; B=0;
do J=10320. to 10331.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(10332.)<0 then WELF(5)=R(10332.);                   else WELF(5)=R(10332.) * B;
end;
if R(10346.)=-4 then PUBLIC(5)=0;                          else do; B=0;
do J=10334. to 10345.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(10346.)<0 then PUBLIC(5)=R(10346.);                  else PUBLIC(5)=R(10346.) * B;
end;
if R(10351.)=-4 then ED(5)=0;                            else ED(5)=R(10351.);
if R(10352.)=-4 then EDSS(5)=0;                          else EDSS(5)=R(10352.);
if R(10354.)=-4 then VET(5)=0;                           else VET(5)=R(10354.);
if R(10358.)=-4 then GIFT(5)=0;                          else GIFT(5)=R(10358.);
if R(10360.)=-4 then OTHER(5)=0;                         else OTHER(5)=R(10360.);
if R(10368.)=-4 then RELWEL(5)=0;                        else RELWEL(5)=R(10368.);
if R(10375.)=-4 then RELREG(5)=0;                        else RELREG(5)=R(10375.);
if R(10779.)=-4 then POVERTY(5)=0;                      else POVERTY(5)=R(10779.);
FAMILY(5)=0;

```

Appendix 2: Total Net Family Income Variable Creation

```

do J=10556. to 10654. BY 7;
  if R(J)<0 ! (R(J)>=33 & R(J)<=36) ! R(J)=45 ! R(J)=46 ! (R(J)>=50 & R(J)<=54) then A=1;
    else FAMILY(5)=FAMILY(5)+1;
  end;
  do I=1 to 5;
    MN=0; FAMINC(I)=0;
    do K=MIL(I), MILS(I), WPS(I), WPSS(I), SEI(I), SEIS(I), UI(I), UIS(I), ALIM(I), CHSP(I), AFDC(I),
      FOOD(I), WELF(I), PUBLIC(I), ED(I), EDSS(I), VET(I), GIFT(I), OTHER(I), RELWEL(I), RELREG(I);
      if K<0 then MN(I,ABS(K))=MN(I,ABS(K))+1; else FAMINC(I)=FAMINC(I) + K;
    end;
    SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
    INCOME(I)=-3;
    if INC(I)=-5 then do; INCOME(I),CPS(I)=-5; end;
    else INC(I)^=-4 then INCOME(I)=INC(I);
    else do;
      if SUMN(I)=0 then INCOME(I)=FAMINC(I);
      else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    end;
    if I>2 then do;
      if FAMILY(I)>0 then do;
        LEVEL(I)=TABLE(I,S(I),FARM(I),FAMILY(I));
        if INC(I)>LEVEL(I) ! FAMINC(I)>LEVEL(I) then CPS(I)=0;
        else if INC(I)>=0 ! (SUMN(I)=0 & INC(I)=-4) then CPS(I)=1;
        else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1; else CPS(I)=-3;
      end;
      else do;
        if INC(I)>TABLE(I,S(I),FARM(I),15) ! FAMINC(I)>TABLE(I,S(I),FARM(I),15) then CPS(I)=0;
        else if (INC(I)>=0 & INC(I)<=TABLE(I,S(I),FARM(I),1)) ! (SUMN(I)=0 & INC(I)=-4 & FAMINC(I)<=
          TABLE(I,S(I),FARM(I),1)) then CPS(I)=1;
        else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1; else CPS(I)=-3;
      end;
      if CPS(I)=-3 & FAMILY(I)>0 & YOUTH(I)> TABLE(I,S(I),FARM(I),FAMILY(I)) then CPS(I)=0;
    end;
    if I=2 then do;
      if INCOME(2)=-5 then CPS(2)=-5;
      else do;
        if FAMILY(2)=1 & INCOME(2)>3778 then CPS(2)=0;
        else if FAMILY(2)=2 & INCOME(2)>4878 then CPS(2)=0;
        else if FAMILY(2)=3 & INCOME(2)>5784 then CPS(2)=0;
        else if FAMILY(2)=4 & INCOME(2)>7412 then CPS(2)=0;
        else if FAMILY(2)=5 & INCOME(2)>8775 then CPS(2)=0;
        else if FAMILY(2)=6 & INCOME(2)>9914 then CPS(2)=0;
        else if FAMILY(2)>6 & INCOME(2)>12280 then CPS(2)=0;
        else if INCOME(2)>12280 then CPS(2)=0;
        else do;
          if INC(2)>=0 & FAMILY(2)>0 then CPS(2)=1;
          else do;
            if SUMN(2)=0 & (R(4027.)=3 ! R(4029.)>1) & FAMILY(2)>0 then CPS(2)=1;
            else if POVERTY(2)>0 then CPS(2)=POVERTY(2)-1; else CPS(2)=-3;
          end;
        end;
      end;
    end;
  end;
  R(2179.)=INCOME(1); R(4060.)=INCOME(2); R(4061.)=POVERTY(2);
  R(6184.10)=INCOME(3); R(6185.)=POVERTY(3); R(8986.)=INCOME(4);
  R(8987.)=POVERTY(4); R(11445.10)=INCOME(5); R(11448.)=POVERTY(5);

```

/ 1984-86 */*

DCL	1 FAMILY_INCOME (6:8),	5 AFDC,	5 ALIM,
	5 CHSP,	5 CPS,	5 ED,
	5 EDSS,	5 FAMILY,	5 FAMINC,
	5 FARM,	5 FOOD,	5 GIFT,
	5 INC,	5 INCOME,	5 LEVEL,
	5 MIL,	5 MILS,	5 MN (3),
	5 OTHER,	5 POVERTY,	5 PUBLIC,
	5 RELREG,	5 RELWEL,	5 S,
	5 SEI,	5 SEIS,	5 SUMN,
	5 TABLE(3,15),	5 UI,	5 UIS,
	5 VET,	5 WELF,	5 WPS,
	5 WPSS;		

/ 1984 VARIABLES */*

```

FAMILY_INCOME=-4;
if R(15196.)=0 then INC(6)=-5; else INC(6)=R(14534.);
TABLE(6,1,1)=5010; TABLE(6,2,1)=6280; TABLE(6,3,1)=6790;
do J=2 to 15; TABLE(6,1,J)=TABLE(6,1,J-1)+1740;
                TABLE(6,2,J)=TABLE(6,2,J-1)+2170; TABLE(6,3,J)=TABLE(6,3,J-1)+1990;
end;
FARM(6)=1; if R(14521.)>4 then FARM(6)=2;
S(6)=1; if STATE84=2 then S(6)=2; else if STATE84=15 then S(6)=3;
LEVEL(6)=-4; if R(14106.)=-4 then MIL(6)=0; else MIL(6)=R(14106.);
if R(14107.)=-4 then WPS(6)=0; else WPS(6)=R(14107.);
if R(14110.)=-4 then SEI(6)=0; else SEI(6)=R(14110.);
if R(14124.)=-4 & R(14125.)=-4 then UI(6)=0;
else if R(14124.)>0 & R(14125.)>0 then UI(6)=R(14124.) * R(14125.);
else if R(14125.)<0 then UI(6)=R(14125.); else UI(6)=-3;
if R(14128.)=-4 then MILS(6)=0; else MILS(6)=R(14128.);
if R(14129.)=-4 then WPSS(6)=0; else WPSS(6)=R(14129.);
if R(14132.)=-4 then SEIS(6)=0; else SEIS(6)=R(14132.);
if R(14146.)=-4 & R(14147.)=-4 then UIS(6)=0;
else if R(14146.)>0 & R(14147.)>0 then UIS(6)=R(14146.) * R(14147.);
else if R(14147.)<0 then UIS(6)=R(14147.); else UIS(6)=-3;
if R(14150.)=-4 then ALIM(6)=0; else ALIM(6)=R(14150.);
if R(14155.)=-4 then CHSP(6)=0; else CHSP(6)=R(14155.);
if R(14171.)=-4 then AFDC(6)=0; else AFDC(6)=R(14171.) * B;
else do; B=0;
do J=14159. to 14170.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(14171.)<0 then AFDC(6)=R(14171.); else AFDC(6)=R(14171.) * B;
end;
if R(14185.)=-4 then FOOD(6)=0; else do; B=0;
do J=14173. to 14184.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(14185.)<0 then FOOD(6)=R(14185.); else FOOD(6)=R(14185.) * B;
end;
if R(14199.)=-4 then WELF(6)=0; else do; B=0;
do J=14187. to 14198.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(14199.)<0 then WELF(6)=R(14199.); else WELF(6)=R(14199.) * B;
end;
if R(14213.)=-4 then PUBLIC(6)=0;

```

Appendix 2: Total Net Family Income Variable Creation

```
else do; B=0;
do J=14201. to 14212.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(14213.)<0 then PUBLIC(6)=R(14213.);
else if B<=0 then PUBLIC(6)=-3;                                else PUBLIC(6)=R(14213.) * B;
end;
if R(14218.)=-4 then ED(6)=0;                                 else ED(6)=R(14218.);
if R(14219.)=-4 then EDSS(6)=0;                               else EDSS(6)=R(14219.);
if R(14221.)=-4 then VET(6)=0;                               else VET(6)=R(14221.);
if R(14225.)=-4 then GIFT(6)=0;                               else GIFT(6)=R(14225.);
if R(14227.)=-4 then OTHER(6)=0;                             else OTHER(6)=R(14227.);
if R(14235.)=-4 then RELWEL(6)=0;                            else RELWEL(6)=R(14235.);
if R(14242.)=-4 then RELREG(6)=0;                           else RELREG(6)=R(14242.);
if R(14536.)=-4 then POVERTY(6)=0;                          else POVERTY(6)=R(14536.);
FAMILY(6)=FAMSIZE;
```

/* To create family size (i.e. FAMSIZE) search thru the household enumeration increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */

Appendix 2: Total Net Family Income Variable Creation

/* 1985 VARIABLES */

```

if R(18902.)=0 then INC(7)=-5;                                else INC(7)=R(18006.);
TABLE(7,1,1)=5180; TABLE(7,2,1)=6500; TABLE(7,3,1)=5970;
do J=2 to 15;                                              TABLE(7,1,J)=TABLE(7,1,J-1)+1810;
                                              TABLE(7,2,J)=TABLE(7,2,J-1)+2260; TABLE(7,3,J)=TABLE(7,3,J-1)+2080;
end;
FARM(7)=1;                                                 if R(17993.)>4 then FARM(7)=2;
S(7)=1; if STATE85=2 then S(7)=2;                           else if STATE85=15 then S(7)=3;
LEVEL(7)=-4; if R(17784.)=-4 then MIL(7)=0;             else MIL(7)=R(17784.);
if R(17785.)=-4 then WPS(7)=0;                            else WPS(7)=R(17785.);
if R(17788.)=-4 then SEI(7)=0;                            else SEI(7)=R(17788.);
if R(17802.)=-4 & R(17803.)=-4 then UI(7)=0;           if R(17802.)>0 & R(17803.)>0 then UI(7)=R(17802.) * R(17803.);
else if R(17803.)<0 then UI(7)=R(17803.);               else UI(7)=-3;
if R(17806.)=-4 then MILS(7)=0;                          else MILS(7)=R(17806.);
if R(17807.)=-4 then WPSS(7)=0;                          else WPSS(7)=R(17807.);
if R(17810.)=-4 then SEIS(7)=0;                          else SEIS(7)=R(17810.);
if R(17824.)=-4 & R(17825.)=-4 then UIS(7)=0;          if R(17824.)>0 & R(17825.)>0 then UIS(7)=R(17824.) * R(17825.);
else if R(17825.)<0 then UIS(7)=R(17825.);            else UIS(7)=-3;
if R(17828.)=-4 then ALIM(7)=0;                         else ALIM(7)=R(17828.);
if R(17833.)=-4 then CHSP(7)=0;                         else CHSP(7)=R(17833.);
if R(17849.)=-4 then AFDC(7)=0;                         else AFDC(7)=R(17849.) * B;
else do; B=0;
do J=17837. to 17848.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(17849.)<0 then AFDC(7)=R(17849.);                  else AFDC(7)=R(17849.) * B;
else if B<=0 then AFDC(7)=-3;
end;
if R(17863.)=-4 then FOOD(7)=0;                         else FOOD(7)=R(17863.) * B;
else do; B=0;
do J=17851. to 17862.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(17863.)<0 then FOOD(7)=R(17863.);                  else FOOD(7)=R(17863.) * B;
else if B<=0 then FOOD(7)=-3;
end;
if R(17877.)=-4 then PUBLIC(7)=0;                        else PUBLIC(7)=R(17877.) * B;
else do; B=0;
do J=17865. to 17876.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(17877.)<0 then PUBLIC(7)=R(17877.);                  else PUBLIC(7)=R(17877.) * B;
else if B<=0 then PUBLIC(7)=-3;
end;
if R(17882.)=-4 then ED(7)=0;                           else ED(7)=R(17882.);
if R(17883.)=-4 then EDSS(7)=0;                         else EDSS(7)=R(17883.);
if R(17885.)=-4 then VET(7)=0;                          else VET(7)=R(17885.);
if R(17887.)=-4 then OTHER(7)=0;                        else OTHER(7)=R(17887.);
if R(17895.)=-4 then RELWEL(7)=0;                      else RELWEL(7)=R(17895.);
if R(17897.)=-4 then RELREG(7)=0;                      else RELREG(7)=R(17897.);
if R(18008.)=-4 then POVERTY(7)=0;                     else POVERTY(7)=R(18008.);
FAMILY(7)=FAMSZ85;

```

/* To create family size (i.e. FAMSZ85) search thru the household enumeration. Increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */

Appendix 2: Total Net Family Income Variable Creation

```

/* 1986 VARIABLES */

if R(22573.)=0 then INC(8)=-5; else INC(8)=R(21622.);
TABLE(8,1,1)=5430; TABLE(8,2,1)=6790; TABLE(8,3,1)=6250;
do J=2 to 15; TABLE(8,1,J)=TABLE(8,1,J-1)+1860;
TABLE(8,2,J)=TABLE(8,2,J-1)+2330; TABLE(8,3,J)=TABLE(8,3,J-1)+2140;
end;
FARM(8)=1; if R(21609.)>4 then FARM(8)=2;
S(8)=1; if STATE86=2 then S(8)=2; else if STATE86=15 then S(8)=3;
LEVEL(8)=-4; if R(21415.)=-4 then MIL(8)=0; else MIL(8)=R(21415.);
if R(21416.)=-4 then WPS(8)=0; else WPS(8)=R(21416.);
if R(21419.)=-4 then SEI(8)=0; else SEI(8)=R(21419.);
if R(21433.)=-4 & R(21434.)=-4 then UI(8)=0; else UI(8)=R(21433.) * R(21434.);
else if R(21433.)>0 & R(21434.)>0 then UI(8)=R(21433.) * R(21434.); else UI(8)=-3;
else if R(21434.)<0 then UI(8)=R(21434.); else UI(8)=-3;
if R(21437.)=-4 then MILS(8)=0; else MILS(8)=R(21437.);
if R(21438.)=-4 then WPSS(8)=0; else WPSS(8)=R(21438.);
if R(21441.)=-4 then SEIS(8)=0; else SEIS(8)=R(21441.);
if R(21455.)=-4 & R(21456.)=-4 then UIS(8)=0; else UIS(8)=R(21455.) * R(21456.);
else if R(21455.)>0 & R(21456.)>0 then UIS(8)=R(21455.) * R(21456.); else UIS(8)=-3;
else if R(21456.)<0 then UIS(8)=R(21456.); else UIS(8)=-3;
if R(21459.)=-4 then ALIM(8)=0; else ALIM(8)=R(21459.);
if R(21464.)=-4 then CHSP(8)=0; else CHSP(8)=R(21464.);
if R(21480.)=-4 then AFDC(8)=0; else AFDC(8)=R(21480.) * B;
else do; B=0;
do J=21468. to 21479.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(21480.)<0 then AFDC(8)=R(21480.); else AFDC(8)=R(21480.) * B;
end;
if R(21494.)=-4 then FOOD(8)=0; else FOOD(8)=R(21494.) * B;
do J=21482. to 21493.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(21494.)<0 then FOOD(8)=R(21494.); else FOOD(8)=R(21494.) * B;
end;
if R(21508.)=-4 then PUBLIC(8)=0; else PUBLIC(8)=R(21508.) * B;
do J=21496. to 21507.; if R(J)>0 & B^=na then B=B+1; else if R(J)>-4 & R(J)<0 then B=na; end;
if R(21508.)<0 then PUBLIC(8)=R(21508.); else PUBLIC(8)=R(21508.) * B;
end;
if R(21513.)=-4 then ED(8)=0; else ED(8)=R(21513.);
if R(21514.)=-4 then EDSS(8)=0; else EDSS(8)=R(21514.);
if R(21516.)=-4 then VET(8)=0; else VET(8)=R(21516.);
if R(21518.)=-4 then OTHER(8)=0; else OTHER(8)=R(21518.);
if R(21526.)=-4 then RELWEL(8)=0; else RELWEL(8)=R(21526.);
if R(21528.)=-4 then RELREG(8)=0; else RELREG(8)=R(21528.);
if R(21624.)=-4 then POVERTY(8)=0; else POVERTY(8)=R(21624.);

FAMILY(8)=FAMSZ86;
/* To create family size (i.e. FAMSZ86) search thru the household enumeration. Increment family size, if the
relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33 & <=36) or =45
or =46 or (>=50 & <=54) */
do I=6 to 8;
MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0;
do K=MIL(I), MILS(I), WPS(I), WPSS(I), SEI(I), SEIS(I), UI(I), UIS(I), ALIM(I), CHSP(I), AFDC(I),
FOOD(I), WELF(I), PUBLIC(I), ED(I), EDSS(I), VET(I), GIFT(I), OTHER(I), RELWEL(I), RELREG(I);
if K>-4 then do;

```

Appendix 2: Total Net Family Income Variable Creation

```
if K<0 then MN(I,ABS(K))=MN(I,ABS(K))+1;           else FAMINC(I)=FAMINC(I) + K;
end;
end;
SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
INCOME(I)=-3;
if INC(I)=-5 then do; INCOME(I)=-5; CPS(I)=-5; end;
else if INC(I)^=-4 then do;
    INCOME(I)=INC(I);
    if INC(I)<0 then do; MN(I,ABS(INC(I)))=1; SUMN(I)=1; end;
end;
else do;
    if SUMN(I)=0 then INCOME(I)=FAMINC(I);
    else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
end;
if FAMILY(I)>0 then do;
    LEVEL(I)=TABLE(I,S(I),FAMILY(I));
    if INC(I)>LEVEL(I) ! FAMINC(I)>LEVEL(I) then CPS(I)=0;
    else if INC(I)>=0 ! (SUMN(I)=0 & INC(I)=-4) then CPS(I)=1;
    else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1;
    else CPS(I)=-3;
end;
else if CPS(I)^=-5 then do;
    if INC(I)>TABLE(I,S(I),15) ! FAMINC(I)>TABLE(I,S(I),15) then CPS(I)=0;
    else if (INC(I)>=0 & INC(I)<=TABLE(I,S(I),1)) ! (SUMN(I)=0 & INC(I)=-4 & FAMINC(I)<=
        TABLE(I,S(I),1)) then CPS(I)=1;
    else if POVERTY(I)>0 then CPS(I)=POVERTY(I)-1;
    else CPS(I)=-3;
end;
end;

INCOME(6)=R(15197.);
CPS(6)=R(15198.);

INCOME(7)=R(18904.);
CPS(7)=R(18905.);

INCOME(8)=R(22575.);
CPS(8)=R(22576.);
```

Appendix 2: Total Net Family Income Variable Creation

/ 1987 VARIABLES */*

DCL	1 FAMILY_INCOME (9), 5 CHSP, 5 EDSS, 5 FAMINC_CENSUS, 5 GIFT, 5 INCOME_CENSUS, 5 MILS, 5 OTHER, 5 RELREG, 5 SEI, 5 SUMN_CENSUS, 5 UIS, 5 WPS,	5 AFDC, 5 CPS, 5 FAMILY, 5 FARM, 5 INC, 5 LEVEL, 5 MN (3), 5 POVERTY, 5 RELWEL, 5 SEIS, 5 TABLE(3,15), 5 VET, 5 WPSS;	5 ALIM, 5 ED, 5 FAMINC, 5 FOOD, 5 INCOME, 5 MIL, 5 MN_CENSUS (3), 5 PUBLIC, 5 S, 5 SUMN, 5 UI, 5 WELF,
-----	---	---	---

/ 1987 POVERTY INCOME GUIDELINES TABLE */*

```

TABLE(9,1,1)=5500; TABLE(9,2,1)=6860; TABLE(9,3,1)=6310;
do J=2 to 15;           TABLE(9,1,J)=TABLE(9,1,J-1)+1900;
                     TABLE(9,2,J)=TABLE(9,2,J-1)+2380; TABLE(9,3,J)=TABLE(9,3,J-1)+2190;
end;
S(9)=1; if STATE87=2 then S(9)=2;
if R(23502.)=-4 then MIL(9)=0;
if R(23503.)=-4 then WPS(9)=0;
if R(23506.)=-4 then SEI(9)=0;
if R(23520.)=-4 & R(23521.)=-4 then UI(9)=0;
else if R(23520.)>0 & R(23521.)>0 then UI(9)=R(23520.) * R(23521.);
else if R(23521.)<0 then UI(9)=R(23521.);
if R(23524.)=-4 then MILS(9)=0;
if R(23525.)=-4 then WPSS(9)=0;
if R(23528.)=-4 then SEIS(9)=0;
if R(23542.)=-4 & R(23543.)=-4 then UIS(9)=0;
else if R(23542.)>0 & R(23543.)>0 then UIS(9)=R(23542.) * R(23543.);
else if R(23543.)<0 then UIS(9)=R(23543.);
if R(23546.)=-4 then ALIM(9)=0;
if R(23551.)=-4 then CHSP(9)=0;
if R(23567.)=-4 then AFDC(9)=0;
else do; B=0;
      if R(23555.)>0 & B^=na then B=B+1;
      if R(23556.)>0 & B^=na then B=B+1;
      if R(23557.)>0 & B^=na then B=B+1;
      if R(23558.)>0 & B^=na then B=B+1;
      if R(23559.)>0 & B^=na then B=B+1;
      if R(23560.)>0 & B^=na then B=B+1;
      if R(23561.)>0 & B^=na then B=B+1;
      if R(23562.)>0 & B^=na then B=B+1;
      if R(23563.)>0 & B^=na then B=B+1;
      if R(23564.)>0 & B^=na then B=B+1;
      if R(23565.)>0 & B^=na then B=B+1;
      if R(23566.)>0 & B^=na then B=B+1;
      if R(23567.)<0 then AFDC(9)=R(23567.); else if B<=0 then AFDC(9)=-3; else AFDC(9)=R(23567.) * B;
end;
if R(23581.)=-4 then FOOD(9)=0;
else do; B=0;
      if R(23569.)>0 & B^=na then B=B+1;
      if R(23570.)>0 & B^=na then B=B+1;
      if R(23571.)>0 & B^=na then B=B+1;
      if R(23572.)>0 & B^=na then B=B+1;
      else if R(23569.)>-4 & R(23569.)<0 then B=na;
      else if R(23570.)>-4 & R(23570.)<0 then B=na;
      else if R(23571.)>-4 & R(23571.)<0 then B=na;
      else if R(23572.)>-4 & R(23572.)<0 then B=na;

```

```

if R(23573.)>0 & B^=na then B=B+1;
if R(23574.)>0 & B^=na then B=B+1;
if R(23575.)>0 & B^=na then B=B+1;
if R(23576.)>0 & B^=na then B=B+1;
if R(23577.)>0 & B^=na then B=B+1;
if R(23578.)>0 & B^=na then B=B+1;
if R(23579.)>0 & B^=na then B=B+1;
if R(23580.)>0 & B^=na then B=B+1;
if R(23581.)<0 then FOOD(9)=R(23581.); else if B<=0 then FOOD(9)=-3; else FOOD(9)=R(23581.) * B;
end;
if R(23595.)=-4 then PUBLIC(9)=0;
else do; B=0;
    if R(23583.)>0 & B^=na then B=B+1;
    if R(23584.)>0 & B^=na then B=B+1;
    if R(23585.)>0 & B^=na then B=B+1;
    if R(23586.)>0 & B^=na then B=B+1;
    if R(23587.)>0 & B^=na then B=B+1;
    if R(23588.)>0 & B^=na then B=B+1;
    if R(23589.)>0 & B^=na then B=B+1;
    if R(23590.)>0 & B^=na then B=B+1;
    if R(23591.)>0 & B^=na then B=B+1;
    if R(23592.)>0 & B^=na then B=B+1;
    if R(23593.)>0 & B^=na then B=B+1;
    if R(23594.)>0 & B^=na then B=B+1;
    if R(23595.)<0 then PUBLIC(9)=R(23595.); else if B<=0 then PUBLIC(9)=-3; else PUBLIC(9)=R(23595.)*B;
end;
if R(23600.)=-4 then ED(9)=0;
if R(23601.)=-4 then EDSS(9)=0;
if R(23603.)=-4 then VET(9)=0;
if R(23605.)=-4 then OTHER(9)=0;
if VET(9)>0 & OTHER(9)=VET(9) then OTHER(9)=0;
if R(23612.)=-4 then RELWEL(9)=0;
if R(23614.)=-4 then RELREG(9)=0;
FAMILY(9)=FAMSZ87; /* To create family size (i.e. FAMSZ87) search thru the household enumeration.
Increment family size if the relationship to the youth is a relative. Do not increase family size if the code is <0
or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=9;
    MN(I,1)=0; MN(I,2)=0; MN(I,3)=0;
    MN_CENSUS(I,1)=0; MN_CENSUS(I,2)=0; MN_CENSUS(I,3)=0;
    FAMINC(I)=0; FAMINC_CENSUS(I)=0;
    INCOME(I)=-3; INCOME_CENSUS(I)=-3;
    COMPONENT(1)=MIL(I);
    COMPONENT(3)=WPS(I);
    COMPONENT(5)=SEI(I);
    COMPONENT(7)=UI(I);
    COMPONENT(9)=ALIM(I);
    COMPONENT(11)=AFDC(I);
    COMPONENT(13)=ED(I);
    COMPONENT(15)=VET(I);
    COMPONENT(17)=RELWEL(I);
    COMPONENT(19)=FOOD(I);
    if WEIGHT(I)=0 then do;
        INCOME(I)=-5; INCOME_CENSUS(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
    end;
    else do;
        do K=1 to 19;
            if R(23573.)>-4 & R(23573.)<0 then B=na;
            else if R(23574.)>-4 & R(23574.)<0 then B=na;
            else if R(23575.)>-4 & R(23575.)<0 then B=na;
            else if R(23576.)>-4 & R(23576.)<0 then B=na;
            else if R(23577.)>-4 & R(23577.)<0 then B=na;
            else if R(23578.)>-4 & R(23578.)<0 then B=na;
            else if R(23579.)>-4 & R(23579.)<0 then B=na;
            else if R(23580.)>-4 & R(23580.)<0 then B=na;
            else if R(23583.)>-4 & R(23583.)<0 then B=na;
            else if R(23584.)>-4 & R(23584.)<0 then B=na;
            else if R(23585.)>-4 & R(23585.)<0 then B=na;
            else if R(23586.)>-4 & R(23586.)<0 then B=na;
            else if R(23587.)>-4 & R(23587.)<0 then B=na;
            else if R(23588.)>-4 & R(23588.)<0 then B=na;
            else if R(23589.)>-4 & R(23589.)<0 then B=na;
            else if R(23590.)>-4 & R(23590.)<0 then B=na;
            else if R(23591.)>-4 & R(23591.)<0 then B=na;
            else if R(23592.)>-4 & R(23592.)<0 then B=na;
            else if R(23593.)>-4 & R(23593.)<0 then B=na;
            else if R(23594.)>-4 & R(23594.)<0 then B=na;
            else ED(9)=R(23600.);
            else EDSS(9)=R(23601.);
            else VET(9)=R(23603.);
            else OTHER(9)=R(23605.);
            else RELWEL(9)=R(23612.);
            else RELREG(9)=R(23614.);

COMPONENT(2)=MILS(I);
COMPONENT(4)=WPSS(I);
COMPONENT(6)=SEIS(I);
COMPONENT(8)=UIS(I);
COMPONENT(10)=CHSP(I);
COMPONENT(12)=PUBLIC(I);
COMPONENT(14)=EDSS(I);
COMPONENT(16)=OTHER(I);
COMPONENT(18)=RELREG(I);

```

Appendix 2: Total Net Family Income Variable Creation

```
if COMPONENT(K)>-4 then do;
  if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
  else FAMINC(I)=FAMINC(I)+COMPONENT(K);
  end;
if K<19 then do;
  if COMPONENT(K)<0 then MN_CENSUS(I,ABS(COMPONENT(K)))=
    MN_CENSUS(I,ABS(COMPONENT(K)))+1;
  else FAMINC_CENSUS(I)=FAMINC_CENSUS(I)+COMPONENT(K);
  end;
  end;
SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
SUMN_CENSUS(I)=MN_CENSUS(I,1) + MN_CENSUS(I,2) + MN_CENSUS(I,3);
if SUMN(I)=0 then INCOME(I)=FAMINC(I);
else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
if SUMN_CENSUS(I)=0 then INCOME_CENSUS(I)=FAMINC_CENSUS(I);
else do J=1 to 3; if MN_CENSUS(I,J)>0 then INCOME_CENSUS(I)=-J; end;
if FAMILY(I)>0 then do;
  LEVEL(I)=TABLE(I,S(I),FAMILY(I));
  if INCOME(I)>LEVEL(I) then CPS(I)=0;
  else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
  else CPS(I)=-3;
end;
end;
end;

do I=1 to 3;
  if MN(9,I)>0 then MN(9,I)=1;
  if MN_CENSUS(9,I)>0 then MN_CENSUS(9,I)=1;
end;
if SUMN(9)>0 then SUMN(9)=1;
if SUMN_CENSUS(9)>0 then SUMN_CENSUS(9)=1;

INCOME(9)=R(24447.);
INCOME_CENSUS(9)=R(24448.);
CPS(9)=R(24449.);
LEVEL(9)=R(24450.);
```

/* 1988 VARIABLES */

```

DCL           1 FAMILY_INCOME (10),
5 (AFDC, ALIM, CHSP, CPS, ED, EDSS, FAMILY, FAMINC, FOOD, INCOME, LEVEL, MIL, MILS,
MN (3), OTHER, PUBLIC, RELREG, RELWEL, S, SEI, SEIS, SUMN, TABLE(3,15), UI, UIS, VET,
WPS, WPSS) fixed DEC(9);
/* 1988 poverty income guidelines table */ TABLE(10,1,1)=5700; TABLE(10,2,1)=7210; TABLE(10,3,1)=6650;
do J=2 to 15;           TABLE(10,1,J)=TABLE(10,1,J-1)+1960;
                        TABLE(10,2,J)=TABLE(10,2,J-1)+2450; TABLE(10,3,J)=TABLE(10,3,J-1)+2250;
end;
S(10)=1; if STATE88=2 then S(10)=2;           else if STATE88=15 then S(10)=3;
if R(27224.)=-4 then MIL(10)=0;           else MIL(10)=R(27224.);
if R(27225.)=-4 then WPS(10)=0;           else WPS(10)=R(27225.);
if R(27228.)=-4 then SEI(10)=0;           else SEI(10)=R(27228.);
if R(27242.)=-4 & R(27243.)=-4 then UI(10)=0;           else if R(27242.)>0 & R(27243.)>0 then UI(10)=R(27242.) * R(27243.);
else if R(27243.)<0 then UI(10)=R(27243.);           else UI(10)=-3;
if R(27246.)=-4 then MILS(10)=0;           else MILS(10)=R(27246.);
if R(27247.)=-4 then WPSS(10)=0;           else WPSS(10)=R(27247.);
if R(27250.)=-4 then SEIS(10)=0;           else SEIS(10)=R(27250.);
if R(27264.)=-4 & R(27265.)=-4 then UIS(10)=0;           else if R(27264.)>0 & R(27265.)>0 then UIS(10)=R(27264.)*R(27265.);
else if R(27265.)<0 then UIS(10)=R(27265.);           else UIS(10)=-3;
if R(27268.)=-4 then ALIM(10)=0;           else ALIM(10)=R(27268.);
if R(27273.)=-4 then CHSP(10)=0;           else CHSP(10)=R(27273.);
if R(27289.)=-4 then AFDC(10)=0;
else do; B=0;
        if R(27277.)>0 & B^=na then B=B+1;           else if R(27277.)>-4 & R(27277.)<0 then B=na;
        if R(27278.)>0 & B^=na then B=B+1;           else if R(27278.)>-4 & R(27278.)<0 then B=na;
        if R(27279.)>0 & B^=na then B=B+1;           else if R(27279.)>-4 & R(27279.)<0 then B=na;
        if R(27280.)>0 & B^=na then B=B+1;           else if R(27280.)>-4 & R(27280.)<0 then B=na;
        if R(27281.)>0 & B^=na then B=B+1;           else if R(27281.)>-4 & R(27281.)<0 then B=na;
        if R(27282.)>0 & B^=na then B=B+1;           else if R(27282.)>-4 & R(27282.)<0 then B=na;
        if R(27283.)>0 & B^=na then B=B+1;           else if R(27283.)>-4 & R(27283.)<0 then B=na;
        if R(27284.)>0 & B^=na then B=B+1;           else if R(27284.)>-4 & R(27284.)<0 then B=na;
        if R(27285.)>0 & B^=na then B=B+1;           else if R(27285.)>-4 & R(27285.)<0 then B=na;
        if R(27286.)>0 & B^=na then B=B+1;           else if R(27286.)>-4 & R(27286.)<0 then B=na;
        if R(27287.)>0 & B^=na then B=B+1;           else if R(27287.)>-4 & R(27287.)<0 then B=na;
        if R(27288.)>0 & B^=na then B=B+1;           else if R(27288.)>-4 & R(27288.)<0 then B=na;
        if R(27289.)<0 then AFDC(10)=R(27289.); else if B<=0 then AFDC(10)=-3; else AFDC(10)=R(27289.) * B;
end;
if R(27303.)=-4 then FOOD(10)=0;
else do; B=0;
        if R(27291.)>0 & B^=na then B=B+1;           else if R(27291.)>-4 & R(27291.)<0 then B=na;
        if R(27292.)>0 & B^=na then B=B+1;           else if R(27292.)>-4 & R(27292.)<0 then B=na;
        if R(27293.)>0 & B^=na then B=B+1;           else if R(27293.)>-4 & R(27293.)<0 then B=na;
        if R(27294.)>0 & B^=na then B=B+1;           else if R(27294.)>-4 & R(27294.)<0 then B=na;
        if R(27295.)>0 & B^=na then B=B+1;           else if R(27295.)>-4 & R(27295.)<0 then B=na;
        if R(27296.)>0 & B^=na then B=B+1;           else if R(27296.)>-4 & R(27296.)<0 then B=na;
        if R(27297.)>0 & B^=na then B=B+1;           else if R(27297.)>-4 & R(27297.)<0 then B=na;
        if R(27298.)>0 & B^=na then B=B+1;           else if R(27298.)>-4 & R(27298.)<0 then B=na;
        if R(27299.)>0 & B^=na then B=B+1;           else if R(27299.)>-4 & R(27299.)<0 then B=na;
        if R(27300.)>0 & B^=na then B=B+1;           else if R(27300.)>-4 & R(27300.)<0 then B=na;
        if R(27301.)>0 & B^=na then B=B+1;           else if R(27301.)>-4 & R(27301.)<0 then B=na;
        if R(27302.)>0 & B^=na then B=B+1;           else if R(27302.)>-4 & R(27302.)<0 then B=na;
        if R(27303.)<0 then FOOD(10)=R(27303.); else if B<=0 then FOOD(10)=-3; else FOOD(10)=R(27303.) * B;
end;
if R(27317.)=-4 then PUBLIC(10)=0;

```

Appendix 2: Total Net Family Income Variable Creation

```

else do; B=0;
if R(27305.)>0 & B^=na then B=B+1;
if R(27306.)>0 & B^=na then B=B+1;
if R(27307.)>0 & B^=na then B=B+1;
if R(27308.)>0 & B^=na then B=B+1;
if R(27309.)>0 & B^=na then B=B+1;
if R(27310.)>0 & B^=na then B=B+1;
if R(27311.)>0 & B^=na then B=B+1;
if R(27312.)>0 & B^=na then B=B+1;
if R(27313.)>0 & B^=na then B=B+1;
if R(27314.)>0 & B^=na then B=B+1;
if R(27315.)>0 & B^=na then B=B+1;
if R(27316.)>0 & B^=na then B=B+1;
if R(27317.)<0 then PUBLIC(10)=R(27317.); else if B<=0 then PUBLIC(10)=-3; else PUBLIC(10)=R(27317.)*B;
end;
if R(27322.)=-4 then ED(10)=0;
if R(27323.)=-4 then EDSS(10)=0;
if R(27325.)=-4 then VET(10)=0;
if R(27332.)=-4 then OTHER(10)=0;
if VET(10)>0 & OTHER(10)=VET(10) then OTHER(10)=0;
if R(27339.)=-4 then RELWEL(10)=0;
if R(27341.)=-4 then RELREG(10)=0;
FAMILY(10)=FAMSZ88; /* To create family size (i.e. FAMSZ88) search thru the household enumeration.
Increment family size if the relationship to the youth is a relative. Do not increase family size if the code is <0
or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=10;
  MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
  COMPONENT(1)=MIL(I);           COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
  COMPONENT(4)=WPSS(I);         COMPONENT(5)=SEI(I);          COMPONENT(6)=SEIS(I);
  COMPONENT(7)=UI(I);           COMPONENT(8)=UIS(I);          COMPONENT(9)=ALIM(I);
  COMPONENT(10)=CHSP(I);        COMPONENT(11)=AFDC(I);        COMPONENT(12)=PUBLIC(I);
  COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);        COMPONENT(15)=VET(I);
  COMPONENT(16)=OTHER(I);       COMPONENT(17)=RELWEL(I);
  COMPONENT(18)=RELREG(I);      COMPONENT(19)=FOOD(I);
  if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
  else do;
    do K=1 to 19;
      if COMPONENT(K)>-4 then do;
        if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
        else FAMINC(I)=FAMINC(I)+COMPONENT(K);
      end;
    end;
    SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
    if SUMN(I)=0 then INCOME(I)=FAMINC(I);
    else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    if FAMILY(I)>0 then do;
      LEVEL(I)=TABLE(I,S(I),FAMILY(I));
      if INCOME(I)>LEVEL(I) then CPS(I)=0;
      else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
      else CPS(I)=-3;
    end;
  end;
  do I=1 to 3; if MN(10,I)>0 then MN(10,I)=1; end;
  if SUMN(10)>0 then SUMN(10)=1;
INCOME(10)=R(28702.);           CPS(10)=R(28704.);           LEVEL(10)=R(28705.);


```

/ 1989 VARIABLES */*

```

DCL           1 FAMILY_INCOME (11),
5 (AFDC, ALIM, CHSP, CPS, ED, EDSS, FAMILY, FAMINC, FOOD, INCOME, LEVEL, MIL, MILS,
MN (3), OTHER, PUBLIC, RELREG, RELWEL, S, SEI, SEIS, SUMN, TABLE(3,15), UI, UIS, VET,
WPS, WPSS) fixed DEC(9);
/* 1989 poverty income guidelines table */ TABLE(11,1,1)=5980; TABLE(11,2,1)=7480; TABLE(11,3,1)=6870;
do J=2 to 15;
    TABLE(11,1,J)=TABLE(11,1,J-1)+2040;
    TABLE(11,2,J)=TABLE(11,2,J-1)+2550; TABLE(11,3,J)=TABLE(11,3,J-1)+2350;
end;
S(11)=1; if R(29922.)=2 then S(11)=2; else if R(29922.)=15 then S(11)=3;
if R(29713.)=-4 then MIL(11)=0; else MIL(11)=R(29713.);
if R(29714.)=-4 then WPS(11)=0; else WPS(11)=R(29714.);
if R(29717.)=-4 then SEI(11)=0; else SEI(11)=R(29717.);
if R(29731.)=-4 & R(29732.)=-4 then UI(11)=0; else if R(29731.)>0 & R(29732.)>0 then UI(11)=R(29731.) * R(29732.);
else if R(29732.)<0 then UI(11)=R(29732.); else UI(11)=-3;
if R(29735.)=-4 then MILS(11)=0; else MILS(11)=R(29735.);
if R(29736.)=-4 then WPSS(11)=0; else WPSS(11)=R(29736.);
if R(29739.)=-4 then SEIS(11)=0; else SEIS(11)=R(29739.);
if R(29753.)=-4 & R(29754.)=-4 then UIS(11)=0; else if R(29753.)>0 & R(29754.)>0 then UIS(11)=R(29753.) * R(29754.);
else if R(29754.)<0 then UIS(11)=R(29754.); else UIS(11)=-3;
if R(29757.)=-4 then ALIM(11)=0; else ALIM(11)=R(29757.);
if R(29759.)=-4 then CHSP(11)=0; else CHSP(11)=R(29759.);
if R(29773.)=-4 then AFDC(11)=0; else do; B=0;
    if R(29761.)>0 & B^=na then B=B+1; else if R(29761.)>-4 & R(29761.)<0 then B=na;
    if R(29762.)>0 & B^=na then B=B+1; else if R(29762.)>-4 & R(29762.)<0 then B=na;
    if R(29763.)>0 & B^=na then B=B+1; else if R(29763.)>-4 & R(29763.)<0 then B=na;
    if R(29764.)>0 & B^=na then B=B+1; else if R(29764.)>-4 & R(29764.)<0 then B=na;
    if R(29765.)>0 & B^=na then B=B+1; else if R(29765.)>-4 & R(29765.)<0 then B=na;
    if R(29766.)>0 & B^=na then B=B+1; else if R(29766.)>-4 & R(29766.)<0 then B=na;
    if R(29767.)>0 & B^=na then B=B+1; else if R(29767.)>-4 & R(29767.)<0 then B=na;
    if R(29768.)>0 & B^=na then B=B+1; else if R(29768.)>-4 & R(29768.)<0 then B=na;
    if R(29769.)>0 & B^=na then B=B+1; else if R(29769.)>-4 & R(29769.)<0 then B=na;
    if R(29770.)>0 & B^=na then B=B+1; else if R(29770.)>-4 & R(29770.)<0 then B=na;
    if R(29771.)>0 & B^=na then B=B+1; else if R(29771.)>-4 & R(29771.)<0 then B=na;
    if R(29772.)>0 & B^=na then B=B+1; else if R(29772.)>-4 & R(29772.)<0 then B=na;
    if R(29773.)<0 then AFDC(11)=R(29773.); else if B<=0 then AFDC(11)=-3; else AFDC(11)=R(29773.) * B;
end;
if R(29787.)=-4 then FOOD(11)=0; else if R(29775.)>-4 & R(29775.)<0 then B=na;
else do; B=0;
    if R(29775.)>0 & B^=na then B=B+1; else if R(29775.)>-4 & R(29775.)<0 then B=na;
    if R(29776.)>0 & B^=na then B=B+1; else if R(29776.)>-4 & R(29776.)<0 then B=na;
    if R(29777.)>0 & B^=na then B=B+1; else if R(29777.)>-4 & R(29777.)<0 then B=na;
    if R(29778.)>0 & B^=na then B=B+1; else if R(29778.)>-4 & R(29778.)<0 then B=na;
    if R(29779.)>0 & B^=na then B=B+1; else if R(29779.)>-4 & R(29779.)<0 then B=na;
    if R(29780.)>0 & B^=na then B=B+1; else if R(29780.)>-4 & R(29780.)<0 then B=na;
    if R(29781.)>0 & B^=na then B=B+1; else if R(29781.)>-4 & R(29781.)<0 then B=na;
    if R(29782.)>0 & B^=na then B=B+1; else if R(29782.)>-4 & R(29782.)<0 then B=na;
    if R(29783.)>0 & B^=na then B=B+1; else if R(29783.)>-4 & R(29783.)<0 then B=na;
    if R(29784.)>0 & B^=na then B=B+1; else if R(29784.)>-4 & R(29784.)<0 then B=na;
    if R(29785.)>0 & B^=na then B=B+1; else if R(29785.)>-4 & R(29785.)<0 then B=na;
    if R(29786.)>0 & B^=na then B=B+1; else if R(29786.)>-4 & R(29786.)<0 then B=na;
    if R(29787.)<0 then FOOD(11)=R(29787.); else if B<=0 then FOOD(11)=-3; else FOOD(11)=R(29787.) * B;
end;
if R(29801.)=-4 then PUBLIC(11)=0;

```

Appendix 2: Total Net Family Income Variable Creation

```

else do; B=0;
if R(29789.)>0 & B^=na then B=B+1;
if R(29790.)>0 & B^=na then B=B+1;
if R(29791.)>0 & B^=na then B=B+1;
if R(29792.)>0 & B^=na then B=B+1;
if R(29793.)>0 & B^=na then B=B+1;
if R(29794.)>0 & B^=na then B=B+1;
if R(29795.)>0 & B^=na then B=B+1;
if R(29796.)>0 & B^=na then B=B+1;
if R(29797.)>0 & B^=na then B=B+1;
if R(29798.)>0 & B^=na then B=B+1;
if R(29799.)>0 & B^=na then B=B+1;
if R(29800.)>0 & B^=na then B=B+1;
if R(29801.)<0 then PUBLIC(11)=R(29801.); else if B<=0 then PUBLIC(11)=-3; else PUBLIC(11)=R(29801.)*B;
end;
if R(29806.)=-4 then ED(11)=0;
if R(29807.)=-4 then EDSS(11)=0;
if R(29809.)=-4 then VET(11)=0;
if R(29813.)=-4 then OTHER(11)=0;
if VET(11)>0 & OTHER(11)=VET(11) then OTHER(11)=0;
if R(29820.)=-4 then RELWEL(11)=0;
if R(29822.)=-4 then RELREG(11)=0;
FAMILY(11)=FAMSZ89; /* To create family size (i.e. FAMSZ89) search thru the household enumeration.
Increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is
<0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=11;
  MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
  COMPONENT(1)=MIL(I); COMPONENT(2)=MILS(I); COMPONENT(3)=WPS(I);
  COMPONENT(4)=WPSS(I); COMPONENT(5)=SEI(I); COMPONENT(6)=SEIS(I);
  COMPONENT(7)=UI(I); COMPONENT(8)=UIS(I); COMPONENT(9)=ALIM(I);
  COMPONENT(10)=CHSP(I); COMPONENT(11)=AFDC(I); COMPONENT(12)=PUBLIC(I);
  COMPONENT(13)=ED(I); COMPONENT(14)=EDSS(I); COMPONENT(15)=VET(I);
  COMPONENT(16)=OTHER(I); COMPONENT(17)=RELWEL(I);
  COMPONENT(18)=RELREG(I); COMPONENT(19)=FOOD(I);
  if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
  else do;
    do K=1 to 19;
      if COMPONENT(K)>-4 then do;
        if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
        else FAMINC(I)=FAMINC(I)+COMPONENT(K);
      end;
    end;
    SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
    if SUMN(I)=0 then INCOME(I)=FAMINC(I);
    else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    if FAMILY(I)>0 then do; LEVEL(I)=TABLE(I,S(I),FAMILY(I));
      if INCOME(I)>LEVEL(I) then CPS(I)=0;
      else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
      else CPS(I)=-3;
    end;
  end;
end;
do I=1 to 3; if MN(11,I)>0 then MN(11,I)=1; end;
if SUMN(11)>0 then SUMN(11)=1;

INCOME(11)=R(30740); CPS(11)=R(30741); LEVEL(11)=R(30742);

```

```

/* 1990 VARIABLES */

do I=12; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
do J=1 to 3; MN(I,J)=-4; end;
OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4; SUMN(I)=-4;
do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
end;
TABLE(12,1,1)=6280; TABLE(12,2,1)=7840; TABLE(12,3,1)=7230;
do J=2 to 15; TABLE(12,1,J)=TABLE(12,1,J-1)+2140; TABLE(12,2,J)=TABLE(12,2,J-1)+2680;
TABLE(12,3,J)=TABLE(12,3,J-1)+2460;
end;
S(12)=1; if STATE90=2 then S(12)=2; else if STATE90=15 then S(12)=3;
if R(32793.)=-4 then MIL(12)=0; else MIL(12)=R(32793.);
if R(32794.)=-4 then WPS(12)=0; else WPS(12)=R(32794.);
if R(32797.)=-4 then SEI(12)=0; else SEI(12)=R(32797.);
if R(32811.)=-4 & R(32812.)=-4 then UI(12)=0; else UI(12)=R(32811.) * R(32812.);
else if R(32811.)>0 & R(32812.)>0 then UI(12)=R(32811.) * R(32812.); else UI(12)=-3;
else if R(32812.)<0 then UI(12)=R(32812.); else MILS(12)=R(32815.);
if R(32815.)=-4 then MILS(12)=0; else MILS(12)=R(32815.);
if R(32816.)=-4 then WPSS(12)=0; else WPSS(12)=R(32816.);
if R(32819.)=-4 then SEIS(12)=0; else SEIS(12)=R(32819.);
if R(32833.)=-4 & R(32834.)=-4 then UIS(12)=0; else UIS(12)=R(32833.) * R(32834.);
else if R(32833.)>0 & R(32834.)>0 then UIS(12)=R(32833.) * R(32834.); else UIS(12)=-3;
else if R(32834.)<0 then UIS(12)=R(32834.); else ALIM(12)=R(32837.);
if R(32837.)=-4 then ALIM(12)=0; else ALIM(12)=R(32837.);
if R(32839.)=-4 then CHSP(12)=0; else CHSP(12)=R(32839.);
if R(32853.)=-4 then AFDC(12)=0; else do; B=0;
else if R(32841.)>0 & B^=na then B=B+1; else if R(32841.)>-4 & R(32841.)<0 then B=na;
else if R(32842.)>0 & B^=na then B=B+1; else if R(32842.)>-4 & R(32842.)<0 then B=na;
if R(32843.)>0 & B^=na then B=B+1; else if R(32843.)>-4 & R(32843.)<0 then B=na;
if R(32844.)>0 & B^=na then B=B+1; else if R(32844.)>-4 & R(32844.)<0 then B=na;
if R(32845.)>0 & B^=na then B=B+1; else if R(32845.)>-4 & R(32845.)<0 then B=na;
if R(32846.)>0 & B^=na then B=B+1; else if R(32846.)>-4 & R(32846.)<0 then B=na;
if R(32847.)>0 & B^=na then B=B+1; else if R(32847.)>-4 & R(32847.)<0 then B=na;
if R(32848.)>0 & B^=na then B=B+1; else if R(32848.)>-4 & R(32848.)<0 then B=na;
if R(32849.)>0 & B^=na then B=B+1; else if R(32849.)>-4 & R(32849.)<0 then B=na;
if R(32850.)>0 & B^=na then B=B+1; else if R(32850.)>-4 & R(32850.)<0 then B=na;
if R(32851.)>0 & B^=na then B=B+1; else if R(32851.)>-4 & R(32851.)<0 then B=na;
if R(32852.)>0 & B^=na then B=B+1; else if R(32852.)>-4 & R(32852.)<0 then B=na;
if R(32853.)<0 then AFDC(12)=R(32853.); else if B<=0 then AFDC(12)=-3; else AFDC(12)=R(32853.) * B;
end;
if R(32867.)=-4 then FOOD(12)=0; else if R(32855.)>0 & R(32855.)<0 then B=na;
else do; B=0; else if R(32855.)>-4 & R(32855.)<0 then B=na;
if R(32856.)>0 & B^=na then B=B+1; else if R(32856.)>-4 & R(32856.)<0 then B=na;
if R(32857.)>0 & B^=na then B=B+1; else if R(32857.)>-4 & R(32857.)<0 then B=na;
if R(32858.)>0 & B^=na then B=B+1; else if R(32858.)>-4 & R(32858.)<0 then B=na;
if R(32859.)>0 & B^=na then B=B+1; else if R(32859.)>-4 & R(32859.)<0 then B=na;
if R(32860.)>0 & B^=na then B=B+1; else if R(32860.)>-4 & R(32860.)<0 then B=na;
if R(32861.)>0 & B^=na then B=B+1; else if R(32861.)>-4 & R(32861.)<0 then B=na;
if R(32862.)>0 & B^=na then B=B+1; else if R(32862.)>-4 & R(32862.)<0 then B=na;
if R(32863.)>0 & B^=na then B=B+1; else if R(32863.)>-4 & R(32863.)<0 then B=na;
if R(32864.)>0 & B^=na then B=B+1; else if R(32864.)>-4 & R(32864.)<0 then B=na;
if R(32865.)>0 & B^=na then B=B+1; else if R(32865.)>-4 & R(32865.)<0 then B=na;
if R(32866.)>0 & B^=na then B=B+1; else if R(32866.)>-4 & R(32866.)<0 then B=na;

```

Appendix 2: Total Net Family Income Variable Creation

```

if R(32867.)<0 then FOOD(12)=R(32867.); else if B<=0 then FOOD(12)=-3; else FOOD(12)=R(32867.) * B;
end;
if R(32893.)=-4 then PUBLIC(12)=0;
else do; B=0;
  if R(32881.)>0 & B^=na then B=B+1;
  if R(32882.)>0 & B^=na then B=B+1;
  if R(32883.)>0 & B^=na then B=B+1;
  if R(32884.)>0 & B^=na then B=B+1;
  if R(32885.)>0 & B^=na then B=B+1;
  if R(32886.)>0 & B^=na then B=B+1;
  if R(32887.)>0 & B^=na then B=B+1;
  if R(32888.)>0 & B^=na then B=B+1;
  if R(32889.)>0 & B^=na then B=B+1;
  if R(32890.)>0 & B^=na then B=B+1;
  if R(32891.)>0 & B^=na then B=B+1;
  if R(32892.)>0 & B^=na then B=B+1;
  if R(32893.)<0 then PUBLIC(12)=R(32893.); else if B<=0 then PUBLIC(12)=-3; else PUBLIC(12)=R(32893.)*B;
end;
if R(32900.)=-4 then ED(12)=0;
if R(32901.)=-4 then EDSS(12)=0;
if R(32902.)=-4 then VET(12)=0;
if R(32906.)=-4 then OTHER(12)=0;
if VET(12)>0 & OTHER(12)=VET(12) then OTHER(12)=0;
if R(32916.)=-4 then RELWEL(12)=0;
if R(32918.)=-4 then RELREG(12)=0;
else if R(32881.)>-4 & R(32881.)<0 then B=na;
else if R(32882.)>-4 & R(32882.)<0 then B=na;
else if R(32883.)>-4 & R(32883.)<0 then B=na;
else if R(32884.)>-4 & R(32884.)<0 then B=na;
else if R(32885.)>-4 & R(32885.)<0 then B=na;
else if R(32886.)>-4 & R(32886.)<0 then B=na;
else if R(32887.)>-4 & R(32887.)<0 then B=na;
else if R(32888.)>-4 & R(32888.)<0 then B=na;
else if R(32889.)>-4 & R(32889.)<0 then B=na;
else if R(32890.)>-4 & R(32890.)<0 then B=na;
else if R(32891.)>-4 & R(32891.)<0 then B=na;
else if R(32892.)>-4 & R(32892.)<0 then B=na;
else ED(12)=R(32900.);
else EDSS(12)=R(32901.);
else VET(12)=R(32902.);
else OTHER(12)=R(32906.);
else RELWEL(12)=R(32916.);
else RELREG(12)=R(32918.);

/* HAND EDITS FOR FAMSZ90 */ if ID=2033 then FAMSZ90=1; if ID=3617 then FAMSZ90=1;
FAMILY(12)=FAMSZ90; /* To create family size (i.e. famsz90) search thru the household enumeration. Increment
family size if the relationship to the youth is a relative. Do not increase family size if the code is <0 or (>=33
& <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=12;
  MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
  COMPONENT(1)=MIL(I); COMPONENT(2)=MILS(I); COMPONENT(3)=WPS(I);
  COMPONENT(4)=WPSS(I); COMPONENT(5)=SEI(I); COMPONENT(6)=SEIS(I);
  COMPONENT(7)=UI(I); COMPONENT(8)=UIS(I); COMPONENT(9)=ALIM(I);
  COMPONENT(10)=CHSP(I); COMPONENT(11)=AFDC(I); COMPONENT(12)=PUBLIC(I);
  COMPONENT(13)=ED(I); COMPONENT(14)=EDSS(I); COMPONENT(15)=VET(I);
  COMPONENT(16)=OTHER(I); COMPONENT(17)=RELWEL(I);
  COMPONENT(18)=RELREG(I); COMPONENT(19)=FOOD(I);
  if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
  else do;
    do K=1 to 19;
      if COMPONENT(K)>-4 then do;
        if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
        else FAMINC(I)=FAMINC(I)+COMPONENT(K);
      end;
    end;
    SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
    if SUMN(I)=0 then INCOME(I)=FAMINC(I); else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    if FAMILY(I)>0 then do; LEVEL(I)=TABLE(I,S(I),FAMILY(I)); if INCOME(I)>LEVEL(I) then CPS(I)=0;
      else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1; else CPS(I)=-3;
    end;
  end;
end;
do I=1 to 3; if MN(12,I)>0 then MN(12,I)=1; end; if SUMN(12)>0 then SUMN(12)=1;
INCOME(12)=R(34007.)           CPS(12)=R(34008.)           LEVEL(12)=R(34009.)

```

Appendix 2: Total Net Family Income Variable Creation

```

/* 1991 VARIABLES */

do I=13; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
do J=1 to 3; MN(I,J)=-4; end;
OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4; SUMN(I)=-4;
do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
end;
TABLE(13,1,1)=6620; TABLE(13,2,1)=8290; TABLE(13,3,1)=7610;
do J=2 to 15; TABLE(13,1,J)=TABLE(13,1,J-1)+2260; TABLE(13,2,J)=TABLE(13,2,J-1)+2820;
TABLE(13,3,J)=TABLE(13,3,J-1)+2600;
end;
S(13)=1; if STATE91=2 then S(13)=2; else if STATE91=15 then S(13)=3;
if R(35589.)=-4 then MIL(13)=0; else MIL(13)=R(35589.);
if R(35590.)=-4 then WPS(13)=0; else WPS(13)=R(35590.);
if R(35593.)=-4 then SEI(13)=0; else SEI(13)=R(35593.);
if R(35607.)=-4 & R(35608.)=-4 then UI(13)=0; else if R(35607.)>0 & R(35608.)>0 then UI(13)=R(35607.) * R(35608.);
else if R(35608.)<0 then UI(13)=R(35608.); else UI(13)=-3;
if R(35611.)=-4 then MILS(13)=0; else MILS(13)=R(35611.);
if R(35612.)=-4 then WPSS(13)=0; else WPSS(13)=R(35612.);
if R(35615.)=-4 then SEIS(13)=0; else SEIS(13)=R(35615.);
if R(35629.)=-4 & R(35630.)=-4 then UIS(13)=0; else if R(35629.)>0 & R(35630.)>0 then UIS(13)=R(35629.) * R(35630.);
else if R(35630.)<0 then UIS(13)=R(35630.); else UIS(13)=-3;
if R(35633.)=-4 then ALIM(13)=0; else ALIM(13)=R(35633.);
if R(35635.)=-4 then CHSP(13)=0; else CHSP(13)=R(35635.);
if R(35649.)=-4 then AFDC(13)=0; else do; B=0;
if R(35637.)>0 & B^=na then B=B+1; else if R(35637.)>-4 & R(35637.)<0 then B=na;
if R(35638.)>0 & B^=na then B=B+1; else if R(35638.)>-4 & R(35638.)<0 then B=na;
if R(35639.)>0 & B^=na then B=B+1; else if R(35639.)>-4 & R(35639.)<0 then B=na;
if R(35640.)>0 & B^=na then B=B+1; else if R(35640.)>-4 & R(35640.)<0 then B=na;
if R(35641.)>0 & B^=na then B=B+1; else if R(35641.)>-4 & R(35641.)<0 then B=na;
if R(35642.)>0 & B^=na then B=B+1; else if R(35642.)>-4 & R(35642.)<0 then B=na;
if R(35643.)>0 & B^=na then B=B+1; else if R(35643.)>-4 & R(35643.)<0 then B=na;
if R(35644.)>0 & B^=na then B=B+1; else if R(35644.)>-4 & R(35644.)<0 then B=na;
if R(35645.)>0 & B^=na then B=B+1; else if R(35645.)>-4 & R(35645.)<0 then B=na;
if R(35646.)>0 & B^=na then B=B+1; else if R(35646.)>-4 & R(35646.)<0 then B=na;
if R(35647.)>0 & B^=na then B=B+1; else if R(35647.)>-4 & R(35647.)<0 then B=na;
if R(35648.)>0 & B^=na then B=B+1; else if R(35648.)>-4 & R(35648.)<0 then B=na;
if R(35649.)<0 then AFDC(13)=R(35649.); else if B<=0 then AFDC(13)=-3; else AFDC(13)=R(35649.) * B;
end;
if R(35663.)=-4 then FOOD(13)=0; else do; B=0;
if R(35651.)>0 & B^=na then B=B+1; else if R(35651.)>-4 & R(35651.)<0 then B=na;
if R(35652.)>0 & B^=na then B=B+1; else if R(35652.)>-4 & R(35652.)<0 then B=na;
if R(35653.)>0 & B^=na then B=B+1; else if R(35653.)>-4 & R(35653.)<0 then B=na;
if R(35654.)>0 & B^=na then B=B+1; else if R(35654.)>-4 & R(35654.)<0 then B=na;
if R(35655.)>0 & B^=na then B=B+1; else if R(35655.)>-4 & R(35655.)<0 then B=na;
if R(35656.)>0 & B^=na then B=B+1; else if R(35656.)>-4 & R(35656.)<0 then B=na;
if R(35657.)>0 & B^=na then B=B+1; else if R(35657.)>-4 & R(35657.)<0 then B=na;
if R(35658.)>0 & B^=na then B=B+1; else if R(35658.)>-4 & R(35658.)<0 then B=na;
if R(35659.)>0 & B^=na then B=B+1; else if R(35659.)>-4 & R(35659.)<0 then B=na;
if R(35660.)>0 & B^=na then B=B+1; else if R(35660.)>-4 & R(35660.)<0 then B=na;
if R(35661.)>0 & B^=na then B=B+1; else if R(35661.)>-4 & R(35661.)<0 then B=na;
if R(35662.)>0 & B^=na then B=B+1; else if R(35662.)>-4 & R(35662.)<0 then B=na;

```

Appendix 2: Total Net Family Income Variable Creation

```

if R(35663.)<0 then FOOD(13)=R(35663.); else if B<=0 then FOOD(13)=-3; else FOOD(13)=R(35663.) * B;
end;
if R(35689.)=-4 then PUBLIC(13)=0;
else do; B=0;
    if R(35677.)>0 & B^=na then B=B+1;
    if R(35678.)>0 & B^=na then B=B+1;
    if R(35679.)>0 & B^=na then B=B+1;
    if R(35680.)>0 & B^=na then B=B+1;
    if R(35681.)>0 & B^=na then B=B+1;
    if R(35682.)>0 & B^=na then B=B+1;
    if R(35683.)>0 & B^=na then B=B+1;
    if R(35684.)>0 & B^=na then B=B+1;
    if R(35685.)>0 & B^=na then B=B+1;
    if R(35686.)>0 & B^=na then B=B+1;
    if R(35687.)>0 & B^=na then B=B+1;
    if R(35688.)>0 & B^=na then B=B+1;
    if R(35689.)<0 then PUBLIC(13)=R(35689.); else if B<=0 then PUBLIC(13)=-3; else PUBLIC(13)=R(35689.)*B;
end;
if R(35695.)=-4 then ED(13)=0;
if R(35696.)=-4 then EDSS(13)=0;
if R(35698.)=-4 then VET(13)=0;
if R(35702.)=-4 then OTHER(13)=0;
if VET(13)>0 & OTHER(13)=VET(13) then OTHER(13)=0;
if R(35712.)=-4 then RELWEL(13)=0;
if R(35714.)=-4 then RELREG(13)=0;
FAMILY(13)=FAMSZ91; /* To create family size (i.e. FAMSZ91) search thru the household enumeratio.n
increment family size if the relationship to the youth is a relative. Do not increase family size if the code is <0
or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=13;
    MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
    COMPONENT(1)=MIL(I); COMPONENT(2)=MILS(I);
    COMPONENT(4)=WPSS(I); COMPONENT(5)=SEI(I);
    COMPONENT(7)=UI(I); COMPONENT(8)=UIS(I);
    COMPONENT(10)=CHSP(I); COMPONENT(11)=AFDC(I);
    COMPONENT(13)=ED(I); COMPONENT(14)=EDSS(I);
    COMPONENT(16)=OTHER(I); COMPONENT(17)=RELWEL(I);
    COMPONENT(18)=RELREG(I); COMPONENT(19)=FOOD(I);
    if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
    end;
    else do;
        do K=1 to 19;
            if COMPONENT(K)>-4 then do;
                if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                else FAMINC(I)=FAMINC(I)+COMPONENT(K);
            end;
            end;
        SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3); if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    if FAMILY(I)>0 then do; LEVEL(I)=TABLE(I,S(I),FAMILY(I)); if INCOME(I)>LEVEL(I) then CPS(I)=0;
        else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1; else CPS(I)=-3;
        end;
    end;
    end;
do I=1 to 3; if MN(13,I)>0 then MN(13,I)=1; end;
if SUMN(13)>0 then SUMN(13)=1;
INCOME(13)=R(36561.)           CPS(13)=R(36562.)           LEVEL(13)=R(36563.)

```

/* 1992 VARIABLES */

```

do I=14; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
do J=1 to 3; MN(I,J)=-4; end;
OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4; SUMN(I)=-4;
do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
end;
TABLE(14,1,1)=6810; TABLE(14,2,1)=8500; TABLE(14,3,1)=7830;
do J=2 to 15; TABLE(14,1,J)=TABLE(14,1,J-1)+2380; TABLE(14,2,J)=TABLE(14,2,J-1)+2980;
TABLE(14,3,J)=TABLE(14,3,J-1)+2740;
end;
S(14)=1; if STATE92=2 then S(14)=2; else if STATE92=15 then S(14)=3;
if R(38970.)=-4 then MIL(14)=0; else MIL(14)=R(38970.);
if R(38971.)=-4 then WPS(14)=0; else WPS(14)=R(38971.);
if R(38974.)=-4 then SEI(14)=0; else SEI(14)=R(38974.);
if R(38988.)=-4 & R(38989.)=-4 then UI(14)=0; else if R(38988.)>0 & R(38989.)>0 then UI(14)=R(38988.) * R(38989.);
else if R(38989.)<0 then UI(14)=R(38989.); else UI(14)=-3;
if R(38992.)=-4 then MILS(14)=0; else MILS(14)=R(38992.);
if R(38993.)=-4 then WPSS(14)=0; else WPSS(14)=R(38993.);
if R(38996.)=-4 then SEIS(14)=0; else SEIS(14)=R(38996.);
if R(39010.)=-4 & R(39011.)=-4 then UIS(14)=0; else if R(39010.)>0 & R(39011.)>0 then UIS(14)=R(39010.) * R(39011.);
else if R(39011.)<0 then UIS(14)=R(39011.); else UIS(14)=-3;
if R(39014.)=-4 then ALIM(14)=0; else ALIM(14)=R(39014.);
if R(39016.)=-4 then CHSP(14)=0; else CHSP(14)=R(39016.);
if R(39030.)=-4 then AFDC(14)=0; else do; B=0;
else if R(39018.)>0 & B^=na then B=B+1; else if R(39018.)>-4 & R(39018.)<0 then B=na;
else if R(39019.)>0 & B^=na then B=B+1; else if R(39019.)>-4 & R(39019.)<0 then B=na;
else if R(39020.)>0 & B^=na then B=B+1; else if R(39020.)>-4 & R(39020.)<0 then B=na;
else if R(39021.)>0 & B^=na then B=B+1; else if R(39021.)>-4 & R(39021.)<0 then B=na;
else if R(39022.)>0 & B^=na then B=B+1; else if R(39022.)>-4 & R(39022.)<0 then B=na;
else if R(39023.)>0 & B^=na then B=B+1; else if R(39023.)>-4 & R(39023.)<0 then B=na;
else if R(39024.)>0 & B^=na then B=B+1; else if R(39024.)>-4 & R(39024.)<0 then B=na;
else if R(39025.)>0 & B^=na then B=B+1; else if R(39025.)>-4 & R(39025.)<0 then B=na;
else if R(39026.)>0 & B^=na then B=B+1; else if R(39026.)>-4 & R(39026.)<0 then B=na;
else if R(39027.)>0 & B^=na then B=B+1; else if R(39027.)>-4 & R(39027.)<0 then B=na;
else if R(39028.)>0 & B^=na then B=B+1; else if R(39028.)>-4 & R(39028.)<0 then B=na;
else if R(39029.)>0 & B^=na then B=B+1; else if R(39029.)>-4 & R(39029.)<0 then B=na;
if R(39030.)<0 then AFDC(14)=R(39030.); else if B<=0 then AFDC(14)=-3; else AFDC(14)=R(39030.) * B;
end;
if R(39044.)=-4 then FOOD(14)=0; else if R(39032.)>-4 & R(39032.)<0 then B=na;
else do; B=0; else if R(39033.)>-4 & R(39033.)<0 then B=na;
if R(39032.)>0 & B^=na then B=B+1; else if R(39033.)>-4 & R(39033.)<0 then B=na;
if R(39033.)>0 & B^=na then B=B+1; else if R(39034.)>-4 & R(39034.)<0 then B=na;
if R(39034.)>0 & B^=na then B=B+1; else if R(39035.)>-4 & R(39035.)<0 then B=na;
if R(39035.)>0 & B^=na then B=B+1; else if R(39036.)>-4 & R(39036.)<0 then B=na;
if R(39036.)>0 & B^=na then B=B+1; else if R(39037.)>-4 & R(39037.)<0 then B=na;
if R(39037.)>0 & B^=na then B=B+1; else if R(39038.)>-4 & R(39038.)<0 then B=na;
if R(39038.)>0 & B^=na then B=B+1; else if R(39039.)>-4 & R(39039.)<0 then B=na;
if R(39039.)>0 & B^=na then B=B+1; else if R(39040.)>-4 & R(39040.)<0 then B=na;
if R(39040.)>0 & B^=na then B=B+1; else if R(39041.)>-4 & R(39041.)<0 then B=na;
if R(39041.)>0 & B^=na then B=B+1; else if R(39042.)>-4 & R(39042.)<0 then B=na;
if R(39042.)>0 & B^=na then B=B+1; else if R(39043.)>-4 & R(39043.)<0 then B=na;
if R(39043.)>0 & B^=na then B=B+1;

```

Appendix 2: Total Net Family Income Variable Creation

```

if R(39044.)<0 then FOOD(14)=R(39044.); else if B<=0 then FOOD(14)=-3; else FOOD(14)=R(39044.) * B;
end;
if R(39070.)=-4 then PUBLIC(14)=0;
else do; B=0;
  if R(39058.)>0 & B^=na then B=B+1;
  if R(39059.)>0 & B^=na then B=B+1;
  if R(39060.)>0 & B^=na then B=B+1;
  if R(39061.)>0 & B^=na then B=B+1;
  if R(39062.)>0 & B^=na then B=B+1;
  if R(39063.)>0 & B^=na then B=B+1;
  if R(39064.)>0 & B^=na then B=B+1;
  if R(39065.)>0 & B^=na then B=B+1;
  if R(39066.)>0 & B^=na then B=B+1;
  if R(39067.)>0 & B^=na then B=B+1;
  if R(39068.)>0 & B^=na then B=B+1;
  if R(39069.)>0 & B^=na then B=B+1;
  if R(39070.)<0 then PUBLIC(14)=R(39070.); else if B<=0 then PUBLIC(14)=-3; else PUBLIC(14)=R(39070.)*B;
end;
if R(39076.)=-4 then ED(14)=0; else ED(14)=R(39076.);
if R(39077.)=-4 then EDSS(14)=0; else EDSS(14)=R(39077.);
if R(39079.)=-4 then VET(14)=0; else VET(14)=R(39079.);
if R(39083.)=-4 then OTHER(14)=0; else OTHER(14)=R(39083.);
if VET(14)>0 & OTHER(14)=VET(14) then OTHER(14)=0;
if R(39093.)=-4 then RELWEL(14)=0; else RELWEL(14)=R(39093.);
if R(39095.)=-4 then RELREG(14)=0; else RELREG(14)=R(39095.);

FAMILY(14)=FAMSZ92; /* To create family size (i.e. famsz92) search thru the household enumeration. Increment
family size if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33
& <=36) or =45 or =46 or (>=50 & <=54) */

DCL COMPONENT(19) fixed DEC(9);
do I=13;
  MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
  COMPONENT(1)=MIL(I); COMPONENT(2)=MILS(I);
  COMPONENT(4)=WPSS(I); COMPONENT(5)=SEI(I);
  COMPONENT(7)=UI(I); COMPONENT(8)=UIS(I);
  COMPONENT(10)=CHSP(I); COMPONENT(11)=AFDC(I);
  COMPONENT(13)=ED(I); COMPONENT(14)=EDSS(I);
  COMPONENT(16)=OTHER(I); COMPONENT(17)=RELWEL(I);
  COMPONENT(18)=RELREG(I); COMPONENT(19)=FOOD(I);
  if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
  else do; do K=1 to 19;
    if COMPONENT(K)>-4 then do;
      if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
      else FAMINC(I)=FAMINC(I)+COMPONENT(K);
    end; end;
    SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3); if SUMN(I)=0 then INCOME(I)=FAMINC(I);
    else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=J; end;
  /* income trunc */ if INCOME(I) > 100000 then INCOME(I) = 833745;
  if FAMILY(I)>0 then do;
    LEVEL(I)=TABLE(I,S(I),FAMILY(I)); if INCOME(I)>LEVEL(I) then CPS(I)=0;
    else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1; else CPS(I)=-3;
  end;
  end;
end;
do I=1 to 3; if MN(14,I)>0 then MN(14,I)=1; end;
if SUMN(14)>0 then SUMN(14)=1;

INCOME(14)=R(40066.)          CPS(14)=R(40067.)          LEVEL(14)=R(40068.)

```

/ 1993 VARIABLES */*

```
do I=15; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
do J=1 to 3; MN(I,J)=-4; end;
OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4; SUMN(I)=-4;
do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
end;
```

/ The 1993 Poverty Income Guidelines are applicable for income received in calendar 1992. Contact the Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation, (202) 690-6141, to receive a fax for the Poverty Income Guidelines. Talk to Gordan Fisher or Joan Turek-Brezina. Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8. */*

/ Several variables pertaining to recipiency from various government income programs were computed from a number of different variables in 1993, due to the change to an event history format for data collection. /* these variables are as follows: */*

```
/* WKTOT92U = Total number of weeks R received unemployment compensation in calendar year 1992 */
/* RCTOT92U = Total amount of unemployment compensation R received in calendar year 1992 */
/* WKTOT92S = Total number of weeks R's spouse received unemployment compensation in calendar year 1992 */
/* RCTOT92U = Total amount of unemployment compensation R's spouse received in calendar year 1992 */
/* MOTOT92A = Total number of months R or spouse received AFDC payments in calendar year 1992 */
/* RCTOT92A = Total amount of AFDC R or spouse received in calendar year 1992 */
/* MOTOT92F = Total number of months R or spouse received government food stamps in calendar year 1992 */
/* RCTOT92F = Total amount of government food stamps R or spouse received in calendar year 1992 */
/* MOTOT92W = Total number of months R or spouse received other welfare or public assistance payments */
/* in calendar year 1992 */
/* RCTOT92W = Total amount of other welfare or public assistance payments R or spouse received in */
/* calendar year 1992 */
```

TABLE(15,1,1)=7360; TABLE(15,2,1)=9200; TABLE(15,3,1)=8470;

do J=2 to 15; TABLE(15,1,J)=TABLE(15,1,J-1)+2480; TABLE(15,2,J)=TABLE(15,2,J-1)+3100;
TABLE(15,3,J)=TABLE(15,3,J-1)+2850;

end;

S(15)=1; if STATE93=2 then S(15)=2; else if STATE93=15 then S(15)=3;
if R(42949.)=-4 then MIL(15)=0; else MIL(15)=R(42949.);

if R(42951.)=-4 then WPS(15)=0; else WPS(15)=R(42951.);

if R(42955.)=-4 then SEI(15)=0; else SEI(15)=R(42955.);

if WKTOT92U=-4 & RCTOT92U=-4 then UI(15)=0;

else if WKTOT92U>0 & RCTOT92U>0 then UI(15)=WKTOT92U * RCTOT92U;

else if RCTOT92U<0 then UI(15)=RCTOT92U; else UI(15)=-3;

if R(43142.)=-4 then MILS(15)=0; else MILS(15)=R(43142.);

if R(43144.)=-4 then WPSS(15)=0; else WPSS(15)=R(43144.);

if R(43149.)=-4 then SEIS(15)=0; else SEIS(15)=R(43149.);

if WKTOT92S=-4 & RCTOT92S=-4 then UIS(15)=0;

else if WKTOT92S>0 & RCTOT92S>0 then UIS(15)=WKTOT92S * RCTOT92S;

else if RCTOT92S<0 then UIS(15)=RCTOT92S; else UIS(15)=-3;

if R(43325.)=-4 then ALIM(15)=0; else ALIM(15)=R(43325.);

if R(43341.)=-4 then CHSP(15)=0; else CHSP(15)=R(43341.);

if R(43351.)=-4 then CHSP(15)=0; else CHSP(15)=R(43351.);

if RCTOT92A=-4 then AFDC(15)=0;

else do; if RCTOT92A<0 then AFDC(15)=RCTOT92A;

else if MOTOT92A<=0 then AFDC(15)=-3;

else AFDC(15)=RCTOT92A * MOTOT92A;

end;

if RCTOT92F=-4 then FOOD(15)=0;

else do; if RCTOT92F<0 then FOOD(15)=RCTOT92F;

Appendix 2: Total Net Family Income Variable Creation

```

        else if MOTOT92F<=0 then FOOD(15)=-3;
        else FOOD(15)=RCTOT92F * MOTOT92F;
    end;
    if RCTOT92W=-4 then PUBLIC(15)=0;
    else do;
        if RCTOT92W<0 then PUBLIC(15)=RCTOT92W;
        else if MOTOT92W<=0 then PUBLIC(15)=-3;
        else PUBLIC(15)=RCTOT92W * MOTOT92W;
    end;
    if R(43879.)=-4 then ED(15)=0;                                else ED(15)=R(43879.);
    if R(43881.)=-4 then EDSS(15)=0;                            else EDSS(15)=R(43881.);
    if R(43883.)=-4 then VET(15)=0;                             else VET(15)=R(43883.);
    if R(43888.)=-4 then OTHER(15)=0;                           else OTHER(15)=R(43888.);
    if VET(15)>0 & OTHER(15)=VET(15) then OTHER(15)=0;
    if R(43900.)=-4 then RELWEL(15)=0;                         else RELWEL(15)=R(43900.);
    if R(43903.)=-4 then RELREG(15)=0;                         else RELREG(15)=R(43903.);
FAMILY(15)=R(44176.); /* To create family size (i.e. R(44176.)) Search thru the household enumeration.
    Increment family size if the relationship to the youth is a relative. Do not increase family size if the code is <0
    or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=15;
    MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
    COMPONENT(1)=MIL(I);      COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
    COMPONENT(4)=WPSS(I);    COMPONENT(5)=SEI(I);          COMPONENT(6)=SEIS(I);
    COMPONENT(7)=UI(I);      COMPONENT(8)=UIS(I);         COMPONENT(9)=ALIM(I);
    COMPONENT(10)=CHSP(I);   COMPONENT(11)=AFDC(I);        COMPONENT(12)=PUBLIC(I);
    COMPONENT(13)=ED(I);     COMPONENT(14)=EDSS(I);       COMPONENT(15)=VET(I);
    COMPONENT(16)=OTHER(I);  COMPONENT(17)=RELWEL(I);
    COMPONENT(18)=RELREG(I); COMPONENT(19)=FOOD(I);
    if WEIGHT(I)=0 then do; INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5; end;
    else do;
        do K=1 to 19;
            if COMPONENT(K)>-4 then do;
                if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                else FAMINC(I)=FAMINC(I)+COMPONENT(K);
            end;
            end;
        SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3); if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    end;
/* income trunc */ if INCOME(I) > 100000 then INCOME(I) = 160097;

    if FAMILY(I)>0 then do;
        LEVEL(I)=TABLE(I,S(I),FAMILY(I));
        if INCOME(I)>LEVEL(I) then CPS(I)=0;
        else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
        else CPS(I)=-3;
    end;
    end;
    end;
do I=1 to 3; if MN(15,I)>0 then MN(15,I)=1; end;
if SUMN(15)>0 then SUMN(15)=1;

INCOME(15)=R(44177.)
CPS(15)=R(44178.)
LEVEL(15)=R(44179.)

```

/ 1994 VARIABLES */*

```
do I=16; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
do J=1 to 3; MN(I,J)=-4; end;
OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4; SUMN(I)=-4;
do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
end;
```

/ The 1994 Poverty Income Guidelines are applicable for income received in calendar 1993. Contact the Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation, (202) 690-6141, to receive a fax for the Poverty Income Guidelines. Talk to Gordan Fisher or Joan Turek-Brezina. Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8. */*

/ Several variables pertaining to recipiency from various government income programs were computed from a number of different variables in 1994, due to the change to an event history format for data collection. */*

/ these variables are as follows: */*

```
/* WKTOT93U = Total number of weeks R received unemployment compensation in calendar year 1993 */
/* RCTOT93U = Total amount of unemployment compensation R received in calendar year 1993 */
/* WKTOT93S = Total number of weeks R's spouse received unemployment compensation in calendar year 1993 */
/* RCTOT93S = Total amount of unemployment compensation R's spouse received in calendar year 1993 */
/* MOTOT93A = Total number of months R or spouse received AFDC payments in calendar year 1993 */
/* RCTOT93A = Total amount of AFDC R or spouse received in calendar year 1993 */
/* MOTOT93F = Total number of months R or spouse received government food stamps in calendar year 1993 */
/* RCTOT93F = Total amount of government food stamps R or spouse received in calendar year 1993 */
/* MOTOT93W = Total number of months R or spouse received other welfare or public assistance payments */
/* in calendar year 1993 */
/* RCTOT93W = Total amount of other welfare or public assistance payments R or spouse received in
calendar year 1993 */
```

TABLE(16,1,1)=7360; TABLE(16,2,1)=9200; TABLE(16,3,1)=8470;
do J=2 to 15; TABLE(16,1,J)=TABLE(16,1,J-1)+2480; TABLE(16,2,J)=TABLE(16,2,J-1)+3100;
TABLE(16,3,J)=TABLE(16,3,J-1)+2850;
end;

```
S(16)=1; if STATE94=2 then S(16)=2; else if STATE94=15 then S(16)=3;
if R(49826)=-4 then MIL(16)=0; else MIL(16)=R(49826.);
if R(49828.)=-4 then WPS(16)=0; else WPS(16)=R(49828.);
if R(49832.)=-4 then SEI(16)=0; else SEI(16)=R(49832.);
if WKTOT93U=-4 & RCTOT93U=-4 then UI(16)=0;
else if WKTOT93U>0 & RCTOT93U>0 then UI(16)=WKTOT93U * RCTOT93U;
else if RCTOT93U<0 then UI(16)=RCTOT93U; else UI(16)=-3;
if R(49958.)=-4 then MILS(16)=0; else MILS(16)=R(49958.);
if R(49960.)=-4 then WPSS(16)=0; else WPSS(16)=R(49960.);
if R(49966.)=-4 then SEIS(16)=0; else SEIS(16)=R(49966.);
if WKTOT93S=-4 & RCTOT93S=-4 then UIS(16)=0;
else if WKTOT93S>0 & RCTOT93S>0 then UIS(16)=WKTOT93S * RCTOT93S;
else if RCTOT93S<0 then UIS(16)=RCTOT93S; else UIS(16)=-3;
if R(50096.)=-4 then ALIM(16)=0; else ALIM(16)=R(50096.);
if R(50119.)=-4 then CHSP(16)=0; else CHSP(16)=R(50119.);
if R(50130.)=-4 then CHSP(16)=0; else CHSP(16)=R(50130.);
if RCTOT93A=-4 then AFDC(16)=0;
else do; if RCTOT93A<0 then AFDC(16)=RCTOT93A;
else if MOTOT93A<=0 then AFDC(16)=-3;
else AFDC(16)=RCTOT93A * MOTOT93A;
end;
if RCTOT93F=-4 then FOOD(16)=0;
else do; if RCTOT93F<0 then FOOD(16)=RCTOT93F;
```

Appendix 2: Total Net Family Income Variable Creation

```

        else if MOTOT93F<=0 then FOOD(16)=-3;
        else FOOD(16)=RCTOT93F * MOTOT93F;
end;
if RCTOT93W=-4 then PUBLIC(16)=0;
else do;
    if RCTOT93W<0 then PUBLIC(16)=RCTOT93W;
    else if MOTOT93W<=0 then PUBLIC(16)=-3;
    else PUBLIC(16)=RCTOT93W * MOTOT93W;
end;
if R(50441.)=-4 then ED(16)=0;                                else ED(16)=R(50441.);
if R(50443.)=-4 then EDSS(16)=0;                            else EDSS(16)=R(50443.);
if R(50445.)=-4 then VET(16)=0;                             else VET(16)=R(50445.);
if R(50450.)=-4 then OTHER(16)=0;                           else OTHER(16)=R(50450.);
if VET(16)>0 & OTHER(16)=VET(16) then OTHER(16)=0;
if R(50462.)=-4 then RELWEL(16)=0;                         else RELWEL(16)=R(50462.);
if R(50465.)=-4 then RELREG(16)=0;                         else RELREG(16)=R(50465.);
FAMILY(16)=FAMSZ94; /* To create family size (i.e. FAMSZ94) search thru the household enumeration.
Increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is
<0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=16;
    MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
    COMPONENT(1)=MIL(I);          COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
    COMPONENT(4)=WPSS(I);        COMPONENT(5)=SEI(I);          COMPONENT(6)=SEIS(I);
    COMPONENT(7)=UI(I);          COMPONENT(8)=UIS(I);         COMPONENT(9)=ALIM(I);
    COMPONENT(10)=CHSP(I);       COMPONENT(11)=AFDC(I);       COMPONENT(12)=PUBLIC(I);
    COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);        COMPONENT(15)=VET(I);
    COMPONENT(16)=OTHER(I);      COMPONENT(17)=RELWEL(I);
    COMPONENT(18)=RELREG(I);     COMPONENT(19)=FOOD(I);
    if WEIGHT(I)=0 then do;
        INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
    end;
    else do;
        do K=1 to 19;
            if COMPONENT(K)>-4 then do;
                if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                else FAMINC(I)=FAMINC(I)+COMPONENT(K);
            end;
        end;
        SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
        if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    end;
/* income trunc */ /* if INCOME(I) > 100000 then INCOME(I) = 500000; */
if FAMILY(I)>0 then do;
    LEVEL(I)=TABLE(I,S(I),FAMILY(I));
    if INCOME(I)>LEVEL(I) then CPS(I)=0;
    else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
    else CPS(I)=-3;
end;
end;
end;
do I=1 to 3; if MN(16,I)>0 then MN(16,I)=1; end;
if SUMN(16)>0 then SUMN(16)=1;

INCOME(16)=R(50807.)           CPS(16)=R(50808.)           LEVEL(16)=R(50809.)

```

/ 1996 VARIABLES */*

```
do I=17; AFDC(I)=-4; ALIM(I)=-4; CHSP(I)=-4; CPS(I)=-4; ED(I)=-4; EDSS(I)=-4; FAMILY(I)=-4;
FAMINC(I)=-4; FOOD(I)=-4; INCOME(I)=-4; LEVEL(I)=-4; MIL(I)=-4; MILS(I)=-4;
do J=1 to 3; MN(I,J)=-4; end;
OTHER(I)=-4; PUBLIC(I)=-4; RELREG(I)=-4; RELWEL(I)=-4; S(I)=-4; SEI(I)=-4; SEIS(I)=-4; SUMN(I)=-4;
do J=1 to 3; do K=1 to 15; TABLE(I,J,K)=-4; end; end;
UI(I)=-4; UIS(I)=-4; VET(I)=-4; WPS(I)=-4; WPSS(I)=-4;
end;
```

/ The 1996 Poverty Income Guidelines are applicable for income received in calendar 1995. Contact the Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation, (202) 690-6141, to receive a fax for the Poverty Income Guidelines. Talk to Gordan Fisher or Joan Turek-Brezina. Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8. */*

/ Several variables pertaining to recipiency from various government income programs were computed from a number of different variables in 1994, due to the change to an event history format for data collection. */*

/ these variables are as follows: */*

```
/* WKTOT95U = Total number of weeks R received unemployment compensation in calendar year 1995 */
/* RCTOT95U = Total amount of unemployment compensation R received in calendar year 1995 */
/* WKTOT95S = Total number of weeks R's spouse received unemployment compensation in calendar year 1995 */
/* RCTOT95S = Total amount of unemployment compensation R's spouse received in calendar year 1995 */
/* MOTOT95A = Total number of months R or spouse received AFDC payments in calendar year 1995 */
/* RCTOT95A = Total amount of AFDC R or spouse received in calendar year 1995 */
/* MOTOT95F = Total number of months R or spouse received government food stamps in calendar year 1995 */
/* RCTOT95F = Total amount of government food stamps R or spouse received in calendar year 1995 */
/* MOTOT95W = Total number of months R or spouse received other welfare or public assistance payments */
/* in calendar year 1995 */
/* RCTOT95W = Total amount of other welfare or public assistance payments R or spouse received in
/* calendar year 1995 */
```

TABLE(17,1,1)=7470; TABLE(17,2,1)=9340; TABLE(17,3,1)=8610;

do J=2 to 15; TABLE(17,1,J)=TABLE(17,1,J-1)+2560; TABLE(17,2,J)=TABLE(17,2,J-1)+3200;
TABLE(17,3,J)=TABLE(17,3,J-1)+2940;

end;

```
S(17)=1; if STATE96=2 then S(17)=2; else if STATE96=15 then S(17)=3;
if R(56260.)=-4 then MIL(17)=0; else MIL(17)=R(56260.);
if R(56262.)=-4 then WPS(17)=0; else WPS(17)=R(56262.);
if R(56266.)=-4 then SEI(17)=0; else SEI(17)=R(56266.);
if WKTOT95U=-4 & RCTOT95U=-4 then UI(17)=0;
else if WKTOT95U>0 & RCTOT95U>0 then UI(17)=WKTOT95U * RCTOT95U;
else if RCTOT95U<0 then UI(17)=RCTOT95U; else UI(17)=-3;
if Q13_16=-4 then MILS(17)=0; else MILS(17)=Q13_16;
if R(56508.)=-4 then WPSS(17)=0; else WPSS(17)=R(56508.);
if R(56514.)=-4 then SEIS(17)=0; else SEIS(17)=R(56514.);
if WKTOT95S=-4 & RCTOT95S=-4 then UIS(17)=0;
else if WKTOT95S>0 & RCTOT95S>0 then UIS(17)=WKTOT95S * RCTOT95S;
else if RCTOT95S<0 then UIS(17)=RCTOT95S; else UIS(17)=-3;
if Q13_32=-4 then ALIM(17)=0; else ALIM(17)=Q13_32;
if Q13_33I=-4 then CHSP(17)=0; else CHSP(17)=Q13_33I;
if Q13_33M=-4 then CHSP(17)=0; else CHSP(17)=Q13_33M;
if RCTOT95A=-4 then AFDC(17)=0;
else do; if RCTOT95A<0 then AFDC(17)=RCTOT95A;
else if MOTOT95A<=0 then AFDC(17)=-3;
else AFDC(17)=RCTOT95A * MOTOT95A;
end;
```

```
if RCTOT95F=-4 then FOOD(17)=0;
else do; if RCTOT95F<0 then FOOD(17)=RCTOT95F;
```

Appendix 2: Total Net Family Income Variable Creation

```

        else if MOTOT95F<=0 then FOOD(17)=-3;
        else FOOD(17)=RCTOT95F * MOTOT95F;
end;
if RCTOT95W=-4 then PUBLIC(17)=0;
else do;
    if RCTOT95W<0 then PUBLIC(17)=RCTOT95W;
    else if MOTOT95W<=0 then PUBLIC(17)=-3;
    else PUBLIC(17)=RCTOT95W * MOTOT95W;
end;
if R(57255.)=-4 then ED(17)=0;                                else ED(17)=R(57255.);
if R(57257.)=-4 then EDSS(17)=0;                             else EDSS(17)=R(57257.);
if Q13_70=-4 then VET(17)=0;                               else VET(17)= Q13_70;
if R(57264.)=-4 then OTHER(17)=0;                            else OTHER(17)=R(57264.);
if VET(17)>0 & OTHER(17)=VET(17) then OTHER(17)=0;
if Q13_92=-4 then RELWEL(17)=0;                            else RELWEL(17)= Q13_92;
if Q13_92C=-4 then RELREG(17)=0;                           else RELREG(17)= Q13_92C;
FAMILY(17)=FAMSZ96; /* To create family size (i.e. FAMSZ96) search thru the household enumeration.
Increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is
<0 or (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */
DCL COMPONENT(19) fixed DEC(9);
do I=17;
    MN(I,1)=0; MN(I,2)=0; MN(I,3)=0; FAMINC(I)=0; INCOME(I)=-3;
    COMPONENT(1)=MIL(I);          COMPONENT(2)=MILS(I);          COMPONENT(3)=WPS(I);
    COMPONENT(4)=WPSS(I);        COMPONENT(5)=SEI(I);          COMPONENT(6)=SEIS(I);
    COMPONENT(7)=UI(I);          COMPONENT(8)=UIS(I);         COMPONENT(9)=ALIM(I);
    COMPONENT(10)=CHSP(I);       COMPONENT(11)=AFDC(I);        COMPONENT(12)=PUBLIC(I);
    COMPONENT(13)=ED(I);          COMPONENT(14)=EDSS(I);        COMPONENT(15)=VET(I);
    COMPONENT(16)=OTHER(I);      COMPONENT(17)=RELWEL(I);
    COMPONENT(18)=RELREG(I);     COMPONENT(19)=FOOD(I);
    if WEIGHT(I)=0 then do;
        INCOME(I)=-5; CPS(I)=-5; LEVEL(I)=-5;
    end;
    else do;
        do K=1 to 19;
            if COMPONENT(K)>-4 then do;
                if COMPONENT(K)<0 then MN(I,ABS(COMPONENT(K)))=MN(I,ABS(COMPONENT(K)))+1;
                else FAMINC(I)=FAMINC(I)+COMPONENT(K);
            end;
        end;
        SUMN(I)=MN(I,1) + MN(I,2) + MN(I,3);
        if SUMN(I)=0 then INCOME(I)=FAMINC(I);
        else do J=1 to 3; if MN(I,J)>0 then INCOME(I)=-J; end;
    end;
/* income trunc */ /* if INCOME(I) > 100000 then INCOME(I) = 500000; */
if FAMILY(I)>0 then do;
    LEVEL(I)=TABLE(I,S(I),FAMILY(I));
    if INCOME(I)>LEVEL(I) then CPS(I)=0;
    else if INCOME(I)>= 0 & INCOME(I)<=LEVEL(I) then CPS(I)=1;
    else CPS(I)=-3;
end;
end;
end;
do I=1 to 3; if MN(17,I)>0 then MN(17,I)=1; end;
if SUMN(17)>0 then SUMN(17)=1;

INCOME(17)=R(51660.)           CPS(17)=R(51661.)           LEVEL(17)=R(51662.)

```

Appendix 2: Total Net Family Income Variable Creation

/* 1998 VARIABLES */

/*programming for the 1998 variable TNFI is done in SPSS. Only question numbers are listed.*/

compute AFDC_CUR=-4	compute CHSP_CUR=-4	compute CHSPS_C=-4
compute CPS_CUR=-4	compute ED_CUR=-4	compute EDSS_CUR=-4
compute FAMILY_C=-4	compute FAMINC_C=-4	compute FOOD_CUR=-4
compute INCOME_C=-4	compute LEVEL_C=-4	compute MIL_CUR=-4
compute MILS_CUR=-4	compute MN1701=-4	compute MN1702=-4
compute MN1703=-4	compute OTHER_C=-4	compute SSI_CUR=-4
compute RELREG_C=-4	compute RELWEL_C=-4	compute S_CUR=-4
compute SEI_CUR=-4	compute SEIS_CUR=-4	compute SUMN_CUR=-4
compute UI_CUR=-4	compute UIS_CUR=-4	compute VET_CUR=-4
compute WPS_CUR=-4	compute WPSS_CUR=-4	

/* The 1997 Poverty Income Guidelines are applicable for income received in calendar 1997. Contact the
/* Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation,
/* (202) 690-6141, to receive a fax for the Poverty Income Guidelines. Talk to Gordan Fisher or Joan
/* Turek-Brezina. Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8. */

/* Several variables pertaining to recipiency from various government income programs were computed from */
/* a number of different variables in 1997, due to the change to an event history format for data collection. */
/* these variables are as follows: */
/* WKTOT97U = Total number of weeks R received unemployment compensation in calendar year 1997 */
/* RCTOT97U = Total amount of unemployment compensation R received in calendar year 1997 */
/* WKTOT97S = Total number of weeks R's spouse received unemployment compensation in calendar year 1997 */
/* RCTOT97U = Total amount of unemployment compensation R's spouse received in calendar year 1997 */
/* MOTOT97A = Total number of months R or spouse received AFDC payments in calendar year 1997 */
/* RCTOT97A = Total amount of AFDC R or spouse received in calendar year 1997 */
/* MOTOT97F = Total number of months R or spouse received government food stamps in calendar year 1997 */
/* RCTOT97F = Total amount of government food stamps R or spouse received in calendar year 1997 */
/* MOTOT97W = Total number of months R or spouse received other welfare or public assistance payments */
/* in calendar year 1997 */
/* RCTOT97W = Total amount of other welfare or public assistance payments R or spouse received in */
/* calendar year 1997 */

compute TABLE1=7890	compute TBL1_INC=2720
compute TABLE2=9870	compute TBL2_INC=3400
compute TABLE3=9070	compute TBL3_INC=3130

compute S_CUR=1	do if (STATE98N = '2')
compute S_CUR=2	else if (STATE98N = '15')
compute S_CUR=3	end if
do if (Q13_3 = -4)	compute MIL_CUR=0
else	compute MIL_CUR=Q13_3
end if	
do if (Q13_5 = -4)	compute WPS_CUR=0
else	compute WPS_CUR=Q13_5
end if	
do if (Q13_9 = -4)	compute SEI_CUR=0
else	compute SEI_CUR=Q13_9
end if	
do if (WKTOT97U = -4 and RCTOT97U = -4)	compute UI_CUR=0
else if (WKTOT97U = -4 and RCTOT97U = 0)	compute UI_CUR=0

Appendix 2: Total Net Family Income Variable Creation

```

else if (WKTOT97U > 0 and RCTOT97U > 0)           compute UI_CUR=(WKTOT97U * RCTOT97U)
else if (WKTOT97U < 0)                             compute UI_CUR=WKTOT97U
else if (RCTOT97U < 0)                             compute UI_CUR=RCTOT97U
else                                                 compute UI_CUR=-3
end if

do if (PAFLAGA EQ 1 or Q13_16 = -4)                 compute MILS_CUR=0
else                                                 compute MILS_CUR=Q13_16
end if

do if (PAFLAGA EQ 1 or Q13_18 = -4)                 compute WPSS_CUR=0
else                                                 compute WPSS_CUR=Q13_18
end if

do if (PAFLAGA EQ 1 or Q13_24 = -4)                 compute SEIS_CUR=0
else                                                 compute SEIS_CUR=Q13_24
end if

do if (PAFLAGA EQ 1 or (WKTOT97S = -4 and RCTOT97S = -4))  compute UIS_CUR=0
else if (PAFLAGA EQ 1 or (WKTOT97S = -4 and RCTOT97S = 0))  compute UIS_CUR=0
else if (WKTOT97S > 0 and RCTOT97S > 0)               compute UIS_CUR=(WKTOT97S * RCTOT97S)
else if (WKTOT97S < 0)                                 compute UIS_CUR=WKTOT97S
else if (RCTOT97S < 0)                               compute UIS_CUR=RCTOT97S
else                                                 compute UIS_CUR=-3
end if

do if (Q13_33I = -4)                                compute CHSP_CUR=0
else                                                 compute CHSP_CUR=Q13_33I
end if

do if (PAFLAGA EQ 1 or Q13_33M = -4)                 compute CHSPS_C=0
else                                                 compute CHSPS_C=Q13_33M
end if

do if (RCTOT97A EQ -4 or RCTOT97A EQ 0)             compute AFDC_CUR=0
else if (MOTOT97A EQ -4 or MOTOT97A EQ 0)           compute AFDC_CUR=0
else if (RCTOT97A < 0)                             compute AFDC_CUR=RCTOT97A
else if (MOTOT97A < 0)                            compute AFDC_CUR=MOTOT97A
else if (MOTOT97A <= 0)                           compute AFDC_CUR=-3
else                                                 compute AFDC_CUR=(RCTOT97A * MOTOT97A)
end if

do if (RCTOT97F EQ -4 or RCTOT97F EQ 0)             compute FOOD_CUR=0
else if (MOTOT97F EQ -4 or MOTOT97F EQ 0)           compute FOOD_CUR=0
else if (RCTOT97F < 0)                             compute FOOD_CUR=RCTOT97F
else if (MOTOT97F < 0)                            compute FOOD_CUR=MOTOT97F
else if (MOTOT97F <= 0)                           compute FOOD_CUR=-3
else                                                 compute FOOD_CUR=(RCTOT97F * MOTOT97F)
end if

do if (RCTOT97W EQ -4 or RCTOT97W EQ 0)             compute SSI_CUR=0
else if (MOTOT97W EQ -4 or MOTOT97W EQ 0)           compute SSI_CUR=0
else if (RCTOT97W < 0)                             compute SSI_CUR=RCTOT97W
else if (MOTOT97W < 0)                            compute SSI_CUR=MOTOT97W
else if (MOTOT97W <= 0)                           compute SSI_CUR=-3
else                                                 compute SSI_CUR=(RCTOT97W * MOTOT97W)
end if

```

Appendix 2: Total Net Family Income Variable Creation

```

do if (Q13_66 = -4)           compute ED_CUR=0
else                          compute ED_CUR=Q13_66
end if

do if (PAFLAGA EQ 1 or Q13_68 = -4)  compute EDSS_CUR=0
else                                compute EDSS_CUR=Q13_68
end if

do if (Q13_70A = -4)           compute VET_CUR=0
else                          compute VET_CUR=Q13_70A
end if

do if (Q13_75 = -4)           compute OTHER_C=0
else                          compute OTHER_C=Q13_75
end if

if (VET_CUR > 0 and OTHER_C = VET_CUR) OTHER_C=0

do if (Q13_92 = -4)           compute RELWEL_C=0
else                          compute RELWEL_C=Q13_92
end if

do if (Q13_92C = -4)          compute RELREG_C=0
else                          compute RELREG_C=Q13_92C
end if

compute FAMILY_C=FAMSZ98 /* To create family size (i.e. famsz98) search thru the household enumeration.
Increment family size, if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or (>=33
& <=36) or =45 or =46 or (>=50 & <=54) */

compute MN1701=0               compute MN1702=0
compute MN1703=0               compute FAMINC_C=0
compute INCOME_C=-3            compute COMPO1=MIL_CUR
compute COMPO2=MILS_CUR         compute COMPO3=WPS_CUR
compute COMPO4=WPSS_CUR         compute COMPO5=SEI_CUR
compute COMPO6=SEIS_CUR         compute COMPO7=UL_CUR
compute COMPO8=UIS_CUR          compute COMPO9=CHSP_CUR
compute COMPO10=AFDC_CUR        compute COMPO11=SSI_CUR
compute COMPO12=ED_CUR          compute COMPO13=EDSS_CUR
compute COMPO14=VET_CUR          compute COMPO15=OTHER_C
compute COMPO16=RELWEL_C         compute COMPO17=RELREG_C
compute COMPO18=FOOD_CUR         compute COMPO19=CHSPS_C

do if (WKTOT97U EQ -4 and RCTOT97U GT 0)    compute UI_CUR=-3      compute COMPO7=-3
else if (WKTOT97U LT 0 and WKTOT97U GT -4)   compute UI_CUR=-3      compute COMPO7=-3
end if

do if (WKTOT97S EQ -4 and RCTOT97S GT 0)    compute UIS_CUR=-3     compute COMPO8=-3
else if (WKTOT97S LT 0 and WKTOT97S GT -4)   compute UIS_CUR=-3     compute COMPO8=-3
end if

do if (MOTOT97A EQ -4 and RCTOT97A GT 0)    compute AFDC_CUR=-3    compute COMPO10=-3
else if (MOTOT97A LT 0 and MOTOT97A GT -4)   compute AFDC_CUR=-3    compute COMPO10=-3
end if

do if (MOTOT97F EQ -4 and RCTOT97F GT 0)    compute FOOD_CUR=-3    compute COMPO18=-3

```

Appendix 2: Total Net Family Income Variable Creation

```
else if (MOTOT97F LT 0 and MOTOT97F GT -4)    compute FOOD_CUR=-3  compute COMPO18=-3
end if

do if (MOTOT97W EQ -4 and RCTOT97W GT 0)    compute SSI_CUR=-3      compute COMPO11=-3
else if (MOTOT97W LT 0 and MOTOT97W GT -4)  compute SSI_CUR=-3      compute COMPO11=-3
end if

do repeat COMPO=COMPO1 to COMPO19
. do if (COMPO > -4)
.   do if (COMPO < 0 and COMPO EQ -1)          .  compute MN1701=(MN1701+1)
.   else if (COMPO < 0 and COMPO EQ -2)        .  compute MN1702=(MN1702+1)
.   else if (COMPO < 0 and COMPO EQ -3)        .  compute MN1703=(MN1703+1)
.   else                                         .  compute FAMINC_C=(FAMINC_C + COMPO)
.   end if
. end if
end repeat

do if (SUMN_CUR = 0)
compute INCOME_C=FAMINC_C
else
. do if (MN1703 > 0)   .  compute INCOME_C=-3
. else if (MN1702 > 0) .  compute INCOME_C=-2
. else if (MN1701 > 0) .  compute INCOME_C=-1
. end if
end if

if (PAFLAGA EQ -5) INCOME_C=-5

/* INCOME TRUNC */ /* if (INCOME_C >= 161400) INCOME_C=1022003 */

/* POVERTY STATUS and LEVEL */
do if (FAMILY_C > 0 and S_CUR EQ 1)
compute LEVEL_C=(TABLE1 + (TBL1_INC * (FAMILY_C-1)))
else if (FAMILY_C > 0 and S_CUR EQ 2)
compute LEVEL_C=(TABLE2 + (TBL2_INC * (FAMILY_C-1)))
else if (FAMILY_C > 0 and S_CUR EQ 3)
compute LEVEL_C=(TABLE3 + (TBL3_INC * (FAMILY_C-1)))
end if

do if (INCOME_C > LEVEL_C)                      compute CPS_CUR=0
else if (INCOME_C >= 0 and INCOME_C <= LEVEL_C) compute CPS_CUR=1
else                                              compute CPS_CUR=-3
end if

do repeat MN=MN1701 to MN1703
if (MN > 0) MN=1
end repeat PRINT

if (SUMN_CUR > 0) SUMN_CUR=1
if (PAFLAGA EQ -5) LEVEL_C=-5
if (PAFLAGA EQ -5) CPS_CUR=-5
```

Appendix 2: Total Net Family Income Variable Creation

/ 2000 VARIABLES */*

*/*programming for the 2000 variable TNFI is done in SPSS. Only question numbers are listed.*/*

compute afdc_cur=-4	compute chsp_cur=-4	compute chsps_c=-4	compute cps_cur=-4
compute ed_cur=-4	compute edss_cur=-4	compute family_c=-4	compute faminc_c=-4
compute food_cur=-4	compute income_c=-4	compute level_c=-4	compute mil_cur=-4
compute mils_cur=-4	compute mn1701=-4	compute mn1702=-4	compute mn1703=-4
compute other_c=-4	compute ssi_cur=-4	compute relreg_c=-4	compute relwel_c=-4
compute s_cur=-4	compute sei_cur=-4	compute seis_cur=-4	compute sumn_cur=-4
compute ui_cur=-4	compute uis_cur=-4	compute vet_cur=-4	compute wps_cur=-4
compute wpss_cur=-4	compute wkcmp_c=-4		

/ The 1999 Poverty Income Guidelines are applicable for income received in calendar 1999. Contact the Dept. of Health and Human Services, the Office of the Assistant Secretary for Planning and Evaluation, (202) 690-6141, to receive a fax for the Poverty Income Guidelines. Talk to Gordan Fisher or Joan Turek-Brezina. Or see the Social Security Bulletin, Annual Statistical Supplement, table 3.E8.*

/ Several variables pertaining to recipiency from various government income programs were computed from a number of different variables in 1999, due to the change to an event history format for data collection.*

/ these variables are as follows:*

<i>/* WKTOT99U = Total number of weeks R received unemployment compensation in calendar year 1999</i>	<i>*/</i>
<i>/* RCTOT99U = Total amount of unemployment compensation R received in calendar year 1999</i>	<i>*/</i>
<i>/* WKTOT99S = Total number of weeks R's spouse received unemployment compensation in calendar year 1999</i>	<i>*/</i>
<i>/* RCTOT99S = Total amount of unemployment compensation R's spouse received in calendar year 1999</i>	<i>*/</i>
<i>/* MOTOT99A = Total number of months R or spouse received AFDC payments in calendar year 1999</i>	<i>*/</i>
<i>/* RCTOT99A = Total amount of AFDC R or spouse received in calendar year 1999</i>	<i>*/</i>
<i>/* MOTOT99F = Total number of months R or spouse received government food stamps in calendar year 1999</i>	<i>*/</i>
<i>/* RCTOT99F = Total amount of government food stamps R or spouse received in calendar year 1999</i>	<i>*/</i>
<i>/* MOTOT99W = Total number of months R or spouse received other welfare or public assistance payments</i>	<i>*/</i>
<i>/* in calendar year 1999</i>	<i>*/</i>
<i>/* RCTOT99W = Total amount of other welfare or public assistance payments R or spouse received in</i>	<i>*/</i>
<i>/* calendar year 1999</i>	<i>*/</i>

compute table1=8240	compute tbl1_inc=2820
compute table2=10320	compute tbl2_inc=3520
compute table3=9490	compute tbl3_inc=3240

/ set state of residence by hand for HA and Alaska for movers */*

/ ste_cd is the original code and norcids are listed for all who reported */*

/ a move to HA or Alaska */*

compute s_cur=1

do if ((ste_cd=2) or ((caseid=5671) or (caseid=11881))) compute s_cur=2
else if ((ste_cd=15) or (caseid=1266)) compute s_cur=3

end if

do if (Q13_3 = -4) compute mil_cur=0
else compute mil_cur=Q13_3 end if

do if (Q13_5 = -4) compute wps_cur=0
else compute wps_cur=Q13_5 end if

/ new series to incorporate new farm & bus income qs */ /* norcids are listed for cases with 2 farms or businesses */*

compute fjt_inc=fjt_141
if (caseid eq 3187) fjt_inc=13000

do if (fjt_inc eq -4) compute farm_cur=0

Appendix 2: Total Net Family Income Variable Creation

```

else                                         compute farm_cur=fjt_inc           end if

compute bpjt_inc=bp_141                      if (caseid eq 2410) bpjt_inc=15000
                                                if (caseid eq 6068) bpjt_inc=40000

do if (bpjt_inc eq -4)                         compute bus_cur=0
else                                         compute bus_cur=bpjt_inc           end if

compute fbr_inc=q13_132d                      compute fbr_cur=0
do if (fbr_inc eq -4)                         compute fbr_cur=fbr_inc           end if
else

do if (Q13_9 = -4)                           compute sei_cur=0
else                                         compute sei_cur=Q13_9             end if

if (farm_cur gt 0) sei_cur=(sei_cur+farm_cur)
if (bus_cur gt 0) sei_cur=(sei_cur+bus_cur)
if (fbr_cur gt 0) sei_cur=(sei_cur+fbr_cur)

do if (wktot99u = -4 and rctot99u = -4)      compute ui_cur=0
else if (wktot99u = -4 and rctot99u = 0)     compute ui_cur=0
else if (wktot99u > 0 and rctot99u > 0)    compute ui_cur=rctot99u
else if (wktot99u < 0)                       compute ui_cur=wktot99u
else if (rctot99u < 0)                       compute ui_cur=rctot99u
else                                         compute ui_cur=-3                 end if

do if (paflaga eq 1 or Q13_16 = -4)          compute mils_cur=0
else                                         compute mils_cur=Q13_16            end if

do if (paflaga eq 1 or Q13_18 = -4)          compute wpss_cur=0
else                                         compute wpss_cur=Q13_18            end if

do if (paflaga eq 1 or Q13_24 = -4)          compute seis_cur=0
else                                         compute seis_cur=Q13_24            end if

do if (paflaga eq 1 or (wktot99s = -4 and rctot99s = -4))  compute uis_cur=0
else if (paflaga eq 1 or (wktot99s = -4 and rctot99s = 0))  compute uis_cur=0
else if (wktot99s > 0 and rctot99s > 0)       compute uis_cur=rctot99s
else if (wktot99s < 0)                         compute uis_cur=wktot99s
else if (rctot99s < 0)                         compute uis_cur=rctot99s
else                                         compute uis_cur=-3                 end if

do if (q13_31b = -4)                           compute wkcmp_c=0
else                                         compute wkcmp_c=q13_31b            end if

do if (Q13_33I = -4)                           compute chsp_cur=0
else                                         compute chsp_cur=Q13_33I            end if

do if (paflaga eq 1 or Q13_33M = -4)          compute chspc_c=0
else                                         compute chspc_c=Q13_33M            end if

do if (rctot99a eq -4 or rctot99a eq 0)      compute afdc_cur=0
else if (motot99a eq -4 or motot99a eq 0)    compute afdc_cur=0
else if (rctot99a < 0)                       compute afdc_cur=rctot99a
else if (motot99a < 0)                       compute afdc_cur=motot99a
else if (motot99a <= 0)                      compute afdc_cur=-3
else                                         compute afdc_cur=rctot99a           end if

```

```

do if (rctot99f eq -4 or rctot99f eq 0)           compute food_cur=0
else if (motot99f eq -4 or motot99f eq 0)         compute food_cur=0
else if (rctot99f < 0)                            compute food_cur=rctot99f
else if (motot99f < 0)                            compute food_cur=motot99f
else if (motot99f <= 0)                           compute food_cur=-3
else                                                 compute food_cur=rctot99f           end if

do if (rctot99w eq -4 or rctot99w eq 0)           compute ssi_cur=0
else if (motot99w eq -4 or motot99w eq 0)         compute ssi_cur=0
else if (rctot99w < 0)                            compute ssi_cur=rctot99w
else if (motot99w < 0)                            compute ssi_cur=motot99w
else if (motot99w <= 0)                           compute ssi_cur=-3
else                                                 compute ssi_cur=rctot99w           end if

do if (Q13_66 = -4)                               compute ed_cur=0
else                                                 compute ed_cur=Q13_66             end if

do if (paflaga eq 1 or Q13_68 = -4)              compute edss_cur=0
else                                                compute edss_cur=Q13_68            end if

do if (q13_70 = -4)                               compute vet_cur=0
else                                                 compute vet_cur=q13_70             end if

do if (Q13_75 = -4)                               compute other_c=0
else                                                compute other_c=Q13_75             end if

if (vet_cur > 0 and other_c = vet_cur) other_c=0

do if (Q13_92 = -4)                               compute relwel_c=0
else                                                compute relwel_c=Q13_92             end if

do if (Q13_92C = -4)                             compute relreg_c=0
else                                                compute relreg_c=Q13_92C            end if

compute family_c=famsz00 /* To create family size (i.e. FAMSZ00) search thru the household enumeration. Increment
                           family size, if the relationship to the youth is a relative. Do not increase family size, if the code is <0 or
                           (>=33 & <=36) or =45 or =46 or (>=50 & <=54) */

compute mn1701=0
compute mn1703=0
compute income_c=-3
compute compo2=mils_cur
compute compo4=wpsCur
compute compo6=seis_cur
compute compo8=uis_cur
compute compo10=afdc_cur
compute compo12=ed_cur
compute compo14=vet_cur
compute compo16=relwel_c
compute compo18=food_cur
compute compo20=wkcmp_c

compute mn1702=0
compute faminc_c=0
compute compo1=mil_cur
compute compo3=wps_cur
compute compo5=sei_cur
compute compo7=ui_cur
compute compo9=chsp_cur
compute compo11=ssi_cur
compute compo13=edss_cur
compute compo15=other_c
compute compo17=relreg_c
compute compo19=chsps_c

do if (wktot99u eq -4 and rctot99u gt 0)        compute ui_cur=-3      compute compo7=-3
else if (wktot99u lt 0 and wktot99u gt -4)       compute ui_cur=-3      compute compo7=-3
end if

```

Appendix 2: Total Net Family Income Variable Creation

```

do if (wktot99s eq -4 and rctot99s gt 0)           compute uis_cur=-3          compute compo8=-3
else if (wktot99s lt 0 and wktot99s gt -4)        compute uis_cur=-3          compute compo8=-3
end if

do if (motot99a eq -4 and rctot99a gt 0)           compute afdc_cur=-3         compute compo10=-3
else if (motot99a lt 0 and motot99a gt -4)        compute afdc_cur=-3         compute compo10=-3
end if

do if (motot99f eq -4 and rctot99f gt 0)           compute food_cur=-3         compute compo18=-3
else if (motot99f lt 0 and motot99f gt -4)        compute food_cur=-3         compute compo18=-3
end if

do if (motot99w eq -4 and rctot99w gt 0)           compute ssi_cur=-3          compute compo11=-3
else if (motot99w lt 0 and motot99w gt -4)        compute ssi_cur=-3          compute compo11=-3
end if

do repeat compo=compo1 to compo20
. do if (compo > -4)
.   do if (compo < 0 and compo eq -1)
.     else if (compo < 0 and compo eq -2)
.       else if (compo < 0 and compo eq -3)
.         else
.       end if
.     end if
.   end repeat print

compute sumn_cur=mn1701 + mn1702 + mn1703

do if (sumn_cur = 0)                                compute income_c=faminc_c
else
. do if (mn1703 > 0)
.   else if (mn1702 > 0)
.     else if (mn1701 > 0)                         . end if
end if

if (paflaga eq -5) income_c=-5
if (caseid eq 9373) income_c=-3

/* poverty status and level */
do if (family_c > 0 and s_cur eq 1)               compute level_c=(table1 + (tbl1_inc * (family_c-1)))
else if (family_c > 0 and s_cur eq 2)               compute level_c=(table2 + (tbl2_inc * (family_c-1)))
else if (family_c > 0 and s_cur eq 3)               compute level_c=(table3 + (tbl3_inc * (family_c-1)))
end if

do if (income_c > level_c)                        compute cps_cur=0
else if (income_c >= 0 and income_c <= level_c)  compute cps_cur=1
else                                                 compute cps_cur=-3
                                                    end if

do repeat mn=mn1701 to mn1703
if (mn > 0) mn=1
end repeat print

if (sumn_cur > 0) sumn_cur=1
if (paflaga eq -5) level_c=-5
if (paflaga eq -5) cps_cur=-5

```

FAMILY POVERTY STATUS AND FAMILY POVERTY LEVEL VARIABLES

Family Poverty Status and Family Poverty Level variables are calculated using the Poverty Income Guidelines, updated yearly by the U.S. Department of Health and Human Services. The Poverty Income Guidelines use one person as a base and an increment is added to that figure for each single person increase in family size.

An NLSY79 respondent is determined to be in poverty if the family income for the last calendar year for the family size is below the Poverty Income Guidelines. The family poverty level, available for survey years 1979, 1987 and later survey years, is the actual federal Poverty Income Guideline amount for each NLSY79 respondent based on the family size for that survey year.

Three sets of guidelines exist, one each for 1) the contiguous 48 states, 2) Alaska, and 3) Hawaii. The figures for Alaska and Hawaii apply scaling factors of 1.25 and 1.15 (respectively) of the figure for the contiguous 48 states. These figures mean that the guidelines for Alaska are 1.25 times those for the contiguous 48 states (or 25% greater), and the guidelines for Hawaii are 1.15 times those of the contiguous 48 states (or 15% greater). In addition, a distinction was made between farm and non-farm families for the contiguous 48 states, Alaska, and Hawaii. Poverty Income Guidelines for farm families were defined as 85% of those for non-farm families. This farm and non-farm distinction was officially eliminated after 1982, and that change was reflected in the NLSY79 variable creation.

For survey years 1980 through 1986, CHRR staff projected poverty guidelines used in the NLSY79 household interview schedule and in the creation of poverty status. This was done because the guidelines were issued in February of the survey year, more than a month after all materials had to be printed for the field period.

For survey years 1979 and 1987 and after, changes in the household interview eliminated the need for projected poverty guidelines, and the actual Poverty Income Guidelines were used in the creation of the NLSY79 poverty status. Because of these changes, there is a higher missing rate in the later survey years on the family poverty status variable for these years.

1979 FAMILY POVERTY STATUS CREATION

The 1979 NLSY79 household interview schedules did not use the Poverty Income Guidelines. As a result, when the family poverty status and the family poverty level variables were created in 1990, the actual federal Poverty Income Guidelines were used.

Since the 1979 family poverty status variable was not created until 1990, a modified version of the family poverty status variable creation was used. First, if the created total net family income variable R(2179.) was greater than zero then income was compared to the Poverty Income Guideline amount for the respondent's family size, farm/non-farm distinction, and state of residence. Second, if the total net family income was less than zero, an abbreviated version of income was calculated using selected income variables (primarily earned income variables and other income variables that did not require computation of the number of months that the income was received). This abbreviated income amount was only used to determine if the family was not in poverty. Third, if the family poverty status was not computed based on either of the previous scenarios, then the screener income R(1916.10), if it was greater than zero, was compared to the Poverty Income Guideline amount for the respondent's screener family size, farm/non-farm distinction, and 1979 state of residence. A special flag variable (R2179.30) was created so that users could determine which income value was used to create the poverty status and the poverty level.

1980-2000 FAMILY POVERTY STATUS AND 1988-2000 FAMILY POVERTY LEVEL CREATION

1980-2000 NLSY79 Family Poverty Status and 1988-2000 Family Poverty Level variable creation is contained in the algorithm for Total Net Family Income, which is part of this appendix.

1980-1986 POVERTY INCOME GUIDELINE PROJECTIONS

The following is a brief description of the procedure used by NLSY79 staff to project the Poverty Income Guidelines that were used in the NLSY79 Family Poverty Status and Family Poverty Level variables. All procedures are applied to the various sets of guidelines and the income increments for those guidelines.

From 1980-1986, NLSY79 staff used the Consumer Price Index (CPI) from the two years preceding the actual NLSY79 survey year to project the Poverty Income Guidelines for the NLSY79 survey year. The following steps were taken:

- 1) The average CPI was calculated for each of the two years preceding the NLSY79 survey year (i.e. for the 1982 NLSY79, the average CPI for 1981 and 1980).
- 2) The average CPI for the most recent year was divided by the average CPI for the next most recent year (i.e. for the 1982 NLSY79, the 1981 average CPI was divided by the 1980 average CPI). The result is the most recent year's CPI as a percentage of the next most recent year's CPI (i.e. for the 1982 NLSY79, the 1981 CPI as a percentage of the 1980 CPI). This is a scaling factor for the amount of change in the average CPI between these two years. This number has been greater than 1 since 1979, because the CPI has always risen between years, not declined. The amount that the scaling figure is greater than 1 is the actual percentage of change between years (i.e. a scaling factor of 1.03561 between 1980 and 1981 indicates that the CPI for 1981 was 1.03561 times the CPI for 1980, or 3.561% greater.)
- 3) The Poverty Income Guidelines for the year preceding the actual NLSY79 survey year are multiplied by the scaling factor resulting from the operation in Step 2 (i.e. for the 1982 NLSY79, the 1981 Poverty Income Guidelines are multiplied by the scaling factor).

This step requires an assumption that the degree of change in the CPI between the two years preceding the NLSY79 survey year is a reasonable estimate of the change that will occur between the year immediately previous and the actual NLSY79 survey year. The accuracy of these projections varies. All may be considered reasonable. However, some figures are more precise than others.

POVERTY INCOME GUIDELINE TABLES

Copies of the computed Poverty Income Guidelines 1980-1986 and the federal Poverty Income Guidelines for 1979, 1987 and subsequent years are part of this documentation.

Appendix 2: Total Net Family Income Variable Creation

1978 Poverty Income Guidelines (applicable for survey year 1979)

Non-Farm				Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 3,400	\$ 4,270	\$ 3,930	\$ 2,910	\$ 3,650	\$ 3,350
2	4,500	5,640	5,910	3,840	4,810	4,420
3	5,600	7,010	6,450	4,770	5,970	5,490
4	6,700	8,380	7,710	5,700	7,130	6,560
5	7,800	9,750	8,970	6,630	8,290	7,630
6	8,900	11,120	10,230	7,560	9,450	8,700
Each add'l person	\$ 1,100	\$ 1,370	\$ 1,260	\$ 930	\$ 1,160	\$ 1,070

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

1979 Poverty Income Guidelines (applicable for survey year 1980)

Non-Farm				Farm		
Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$3,770	\$4,740	\$4,350	\$3,220	\$4,030	\$3,690
2	5,000	6,270	5,760	4,260	5,330	4,890
3	6,230	7,800	7,170	5,300	6,630	6,090
4	7,460	9,330	8,580	6,340	7,930	7,290
5	8,690	10,860	9,990	7,380	9,230	8,490
6	9,920	12,390	11,400	8,420	10,530	9,690
7	11,150	13,920	12,810	9,460	11,830	10,890
8	12,380	15,450	14,220	10,500	13,130	12,090
9	13,610	16,980	15,630	11,540	14,430	13,290
10	14,840	18,510	17,040	12,580	15,730	14,490
11	16,070	20,040	18,450	13,620	17,030	15,690
12	17,300	21,570	19,860	14,660	18,330	16,890
13	18,530	23,100	21,270	15,700	19,630	18,090
14	19,760	24,630	22,680	16,740	20,930	19,290
15	20,990	26,160	24,090	17,780	22,230	20,490
16	22,220	27,690	25,500	18,820	23,530	21,690
17	23,450	29,220	26,910	19,860	24,830	22,890
18	24,680	30,750	28,320	20,900	26,130	24,090
19	25,910	32,280	29,730	21,940	27,430	25,290
20	27,140	33,810	31,140	22,980	28,730	26,490
Each add'l person	\$ 1230	\$ 1530	\$ 1410	\$ 1040	\$ 1300	\$ 1200

Appendix 2: Total Net Family Income Variable Creation

1980 Poverty Income Guidelines (applicable for survey year 1981)

Size of family unit	Non-Farm			Farm		
	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 4,320	\$ 5,400	\$ 4,970	\$ 3,690	\$ 4,660	\$ 4,270
2	5,700	7,130	6,560	4,860	6,110	5,610
3	7,080	8,860	8,150	6,030	7,560	6,950
4	8,460	10,590	9,740	7,200	9,010	8,290
5	9,840	12,320	11,330	8,370	10,460	9,630
6	11,220	14,050	12,900	9,540	11,910	10,970
7	12,600	15,780	14,510	10,710	13,360	12,310
8	13,980	17,510	16,100	11,880	14,810	13,650
9	15,360	19,240	17,690	13,050	16,260	14,990
10	16,740	20,970	19,280	14,220	17,710	16,330
11	18,120	22,700	20,870	15,390	19,160	17,670
12	19,500	24,430	22,460	16,560	20,610	19,010
13	20,880	26,160	24,050	17,730	22,060	20,350
14	22,260	27,890	25,640	18,900	23,510	21,690
15	23,640	29,620	27,230	20,070	24,960	23,030
16	25,020	31,350	28,820	21,240	26,410	24,370
17	26,400	33,080	30,410	22,410	27,860	25,710
18	27,780	34,810	32,000	23,580	29,310	27,050
19	29,160	36,540	33,590	24,750	30,760	28,390
20	30,540	38,270	35,180	25,920	32,210	29,730
Each add'l person	\$ 1,380	\$ 1,730	\$ 1,590	\$ 1,170	\$ 1,450	\$ 1,340

Appendix 2: Total Net Family Income Variable Creation

1981 Poverty Income Guidelines (applicable for survey year 1982)

Size of family unit	Non-Farm			Farm		
	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 4,320	\$ 5,400	\$ 4,970	\$ 3,690	\$ 4,660	\$ 4,270
2	5,700	7,130	6,560	4,860	6,110	5,610
3	7,080	8,860	8,150	6,030	7,560	6,950
4	8,460	10,590	9,740	7,200	9,010	8,290
5	9,840	12,320	11,330	8,370	10,460	9,630
6	11,220	14,050	12,900	9,540	11,910	10,970
7	12,600	15,780	14,510	10,710	13,360	12,310
8	13,980	17,510	16,100	11,880	14,810	13,650
9	15,360	19,240	17,690	13,050	16,260	14,990
10	16,740	20,970	19,280	14,220	17,710	16,330
11	18,120	22,700	20,870	15,390	19,160	17,670
12	19,500	24,430	22,460	16,560	20,610	19,010
13	20,880	26,160	24,050	17,730	22,060	20,350
14	22,260	27,890	25,640	18,900	23,510	21,690
15	23,640	29,620	27,230	20,070	24,960	23,030
16	25,020	31,350	28,820	21,240	26,410	24,370
17	26,400	33,080	30,410	22,410	27,860	25,710
18	27,780	34,810	32,000	23,580	29,310	27,050
19	29,160	36,540	33,590	24,750	30,760	28,390
20	30,540	38,270	35,180	25,920	32,210	29,730
Each add'l person	\$ 1,380	\$ 1,730	\$ 1,590	\$ 1,170	\$ 1,450	\$ 1,340

1982 Poverty Income Guidelines (applicable for survey year 1983)

Size of family unit	Non-Farm			Farm		
	Contiguous (48) states and District of Columbia	Alaska	Hawaii	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 4,910	\$ 6,160	\$ 5,670	\$ 4,200	\$ 5,290	\$ 4,850
2	6,530	8,180	7,530	5,570	6,990	6,420
3	8,150	10,200	9,390	6,940	8,690	7,990
4	9,770	12,220	11,250	8,310	10,390	9,560
5	11,390	14,240	13,110	9,680	12,090	11,130
6	13,010	16,260	14,970	11,050	13,790	12,700
7	14,630	18,280	16,830	12,420	15,490	14,270
8	16,250	20,300	18,690	13,790	17,190	15,840
9	17,870	22,320	20,550	15,160	18,890	17,410
10	19,490	24,340	22,410	16,530	20,590	18,980
11	21,110	26,360	24,270	17,900	22,290	20,550
12	22,730	28,380	26,130	19,270	23,990	22,120
13	24,350	30,400	27,990	20,640	25,690	23,690
14	25,970	32,420	29,850	22,010	27,390	25,260
15	27,590	34,440	31,710	23,380	29,090	26,830
Each add'l person	\$ 1,620	\$ 2,020	\$ 1,860	\$ 1,370	\$ 1,700	\$ 1,570

Appendix 2: Total Net Family Income Variable Creation

1983 Poverty Income Guidelines (applicable for survey year 1984)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 5,010	\$ 6,280	\$ 6,790
2	6,750	8,450	7,780
3	8,490	10,620	9,770
4	10,230	12,790	11,760
5	11,970	14,960	13,750
6	13,710	17,130	15,740
7	15,450	19,300	17,730
8	17,190	21,470	19,720
9	18,930	23,640	21,710
10	20,670	25,810	23,700
11	22,410	27,980	25,690
12	24,150	30,150	27,680
13	25,890	32,320	29,670
14	27,630	34,490	31,660
15	29,370	36,660	33,650
Each add'l person	\$ 1,740	\$ 2,170	\$ 1,990

1984 Poverty Income Guidelines (applicable for survey year 1985)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 5,180	\$ 6,500	\$ 5,970
2	6,990	8,760	8,050
3	8,800	11,020	10,130
4	10,610	13,280	12,210
5	12,420	15,540	14,290
6	14,230	17,800	16,370
7	16,040	20,060	18,450
8	17,850	22,320	20,530
9	19,660	24,580	22,610
10	21,470	26,840	24,690
11	23,280	29,100	26,770
12	25,090	31,360	28,850
13	26,900	33,620	30,930
14	28,710	35,880	33,010
15	30,520	38,140	35,090
16	32,330	40,400	37,170
17	34,140	42,660	39,250
18	35,950	44,920	41,330
19	37,760	47,180	43,410
20	39,570	49,440	45,490
Each add'l person	\$ 1,810	\$ 2,260	\$ 2,080

Appendix 2: Total Net Family Income Variable Creation

1985 Poverty Income Guidelines (applicable for survey year 1986)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 5,430	\$ 6,790	\$ 6,250
2	7,290	9,120	8,390
3	9,150	11,450	10,530
4	11,010	13,780	12,670
5	12,870	16,110	14,810
6	14,730	18,440	16,950
7	16,590	20,770	19,090
8	18,450	23,100	21,230
9	20,310	25,430	23,370
10	22,170	27,760	25,510
11	24,030	30,090	27,650
12	25,890	32,420	29,790
13	27,750	34,750	31,930
14	29,610	37,080	34,070
15	31,470	39,410	36,210
16	33,330	41,740	38,350
17	35,190	44,070	40,490
18	37,050	46,400	42,630
19	38,910	48,730	44,770
20	40,770	51,060	46,910
Each add'l person	\$ 1,860	\$ 2,330	\$ 2,140

1986 Poverty Guidelines (applicable for survey year 1987)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 5,550	\$ 6,680	\$ 6,310
2	7,400	9,240	8,500
3	9,300	11,620	10,690
4	11,200	14,000	12,880
5	13,100	16,380	15,070
6	15,000	18,760	17,260
7	16,900	21,140	19,450
8	18,800	23,520	21,640
Each add'l person	\$ 1,900	\$ 2,380	\$ 2,190

Appendix 2: Total Net Family Income Variable Creation

1987 Poverty Guidelines (applicable for survey year 1988)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 5,770	\$ 7,210	\$ 6,650
2	7,730	9,660	8,900
3	9,690	12,110	11,150
4	11,650	14,560	13,400
5	13,610	17,010	15,650
6	15,570	19,460	17,900
7	17,530	21,910	20,150
8	19,490	24,360	22,400
Each add'l person	\$ 1,960	\$ 2,450	\$ 2,250

1988 Poverty Income Guidelines (applicable for survey year 1989)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 5,980	\$ 7,480	\$ 6,870
2	8,020	10,030	9,220
3	10,060	12,580	11,570
4	12,100	15,130	13,920
5	14,140	17,680	16,270
6	16,180	20,230	18,620
7	18,220	22,780	20,970
8	20,260	25,330	23,320
Each add'l person	\$ 2,040	\$ 2,550	\$ 2,350

1989 Poverty Income Guidelines (applicable for survey year 1990)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 6,280	\$ 7,840	\$ 7,230
2	8,420	10,520	9,690
3	10,560	13,200	12,150
4	12,700	15,880	14,610
5	14,840	18,560	17,070
6	16,980	21,240	19,530
7	19,120	23,920	21,990
8	21,260	26,600	24,450
Each add'l person	\$ 2,140	\$ 2,680	\$ 2,460

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

Appendix 2: Total Net Family Income Variable Creation

1990 Poverty Income Guidelines (applicable for survey year 1991)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 6,620	\$ 8,290	\$ 7,610
2	8,880	11,110	10,210
3	11,140	13,930	12,810
4	13,400	16,750	15,410
5	15,660	19,570	18,010
6	17,920	22,390	20,610
7	20,180	25,210	23,210
8	22,440	28,030	25,810
Each add'l person	\$ 2,260	\$ 2,820	\$ 2,600

1991 Poverty Income Guidelines (applicable for survey year 1992)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 6,810	\$ 8,500	\$ 7,830
2	9,190	11,480	10,570
3	11,570	14,460	13,310
4	13,950	17,440	16,050
5	16,330	20,420	18,790
6	18,710	23,400	21,530
7	21,090	26,380	24,270
8	23,470	29,360	27,010
Each add'l person	\$ 2,380	\$ 2,980	\$ 2,740

1992 Poverty Income Guidelines (applicable for survey year 1993)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 7,360	\$ 9,200	\$ 8,470
2	9,840	12,300	11,320
3	12,320	15,400	14,170
4	14,800	18,500	17,020
5	17,280	21,600	19,870
6	19,760	24,700	22,720
7	22,240	27,800	25,570
8	24,720	30,900	28,420
Each add'l person	\$ 2,480	\$ 3,100	\$ 2,850

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

Appendix 2: Total Net Family Income Variable Creation

1993 Poverty Income Guidelines (applicable for survey year 1994)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 7,360	\$ 9,200	\$ 8,470
2	9,840	12,300	11,320
3	12,320	15,400	14,170
4	14,800	18,500	17,020
5	17,280	21,600	19,870
6	19,760	24,700	22,720
7	22,240	27,800	25,570
8	24,720	30,900	28,420
Each add'l person	\$ 2,480	\$ 3,100	\$ 2,850

1995 Poverty Income Guidelines (applicable for survey year 1996)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 7,470	\$ 9,340	\$ 8,610
2	10,030	12,540	11,550
3	12,590	15,740	14,490
4	15,150	18,940	17,430
5	17,710	22,140	20,370
6	20,270	25,340	23,310
7	22,830	28,540	26,250
8	25,390	31,740	29,190
Each add'l person	\$ 2,560	\$ 3,200	\$ 2,940

1997 Poverty Income Guidelines (applicable for survey year 1998)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 7,890	\$ 9,870	\$ 9,070
2	10,610	13,270	12,200
3	13,330	16,670	15,330
4	16,050	20,070	18,460
5	18,770	23,470	21,590
6	21,490	26,870	24,720
7	24,210	30,270	27,850
8	26,930	33,670	30,980
Each add'l person	\$ 2,720	\$ 3,400	\$ 3,130

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

Appendix 2: Total Net Family Income Variable Creation

1999 Poverty Income Guidelines (applicable for survey year 2000)

Size of family unit	Contiguous (48) states and District of Columbia	Alaska	Hawaii
1	\$ 8,240	\$10,320	\$ 9490
2	11,060	13,840	12,730
3	13,880	17,360	15,970
4	16,700	20,880	19,210
5	19,520	24,400	22,450
6	22,340	27,920	25,690
7	25,160	31,440	28,930
8	27,980	34,960	32,170
Each add'l person	\$ 2,820	\$ 3,520	\$ 3,240

Questions about the use of the poverty guidelines in one of the outlying jurisdictions (Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Marshall Islands, the Federated States of Micronesia, the Commonwealth of the North Marianas, and Palau) by a Federal program serving any of those jurisdictions should be referred to the Federal office which is responsible for the individual program involved.

NLSY79 APPENDIX 3:

JOB SATISFACTION MEASURES

JOB SATISFACTION MEASURES

Included in the 1979 through 1982 NLSY79 is a measure of job satisfaction with econometric properties that make it better than the global job satisfaction measure also included in the surveys. The job satisfaction measure is a seven-item scale that has greater variance than the global satisfaction measure. The scale has been tested previously on broad cross-sections of national data.

The following is a list of the reference numbers for the job satisfaction scale variables and the global satisfaction variables for all of the current survey years in which it was used.

Survey Year	Reference # for Scale Items						Reference # for Global Satisfaction
1979	R00489.	R00490.	R00494.	R00496.	R00497.	R00506.	R00508.
1980	R02659.	R02660.	R02664.	R02666.	R02667.	R02676.	R02678.
1981	R04473.	R04474.	R04478.	R04480.	R04481.	R04490.	R04492.
1982	R07034.	R07035.	R07139.	R07041.	R07042.	R07052.	R07065.
1988	R25296.	R25297.	R25302.	R25304.	R25305.		R25329.

This scale is a shortened form of the job satisfaction scales perfected for use in the University of Michigan's Quality of Employment Surveys of 1969, 1973, and 1977. Short forms were developed and documented by Robert Quinn.¹ The Quality of Employment Survey scale was chosen for inclusion in the NLS because it has been analyzed for its reliability across a broad cross-section of workers. It is also much easier to administer for a large multi-purpose survey than are many of the other job satisfaction scales that exist.²

To construct the full seven-item scale, raw scores for each item should be converted to z scores for each respondent. The scores can then be multiplied by 100 to remove decimals and combined to obtain an unweighted average of the seven z scores. The resulting scores for the satisfaction index are either positive or negative numbers that can be interpreted as deviations from the mean for the total sample of respondents in the survey.

NOTES

¹Quinn, R.B. and Mangione, T.W. "Jobsat '72 and Its Kinfolk - A Brief Manual." Chapter 5 in *The 1969-1970 Survey of Working Conditions: Chronicles of an Unfinished Enterprise*. Ann Arbor: Survey Research Center, Institute for Social Research, University of Michigan, 1973.

²Data on the validities of the job satisfaction measures in the Quality of Employment Survey are reported in Mangione, T., "The Validity of Job Satisfaction." Doctoral dissertation, The University of Michigan, 1973. For a review of job satisfaction indicators, see Seashore, S. and Taber, T., "Job Satisfaction Indicators and Their Correlates." *American Behavioral Scientist* 18 (1975), 333-368.

NLSY79 APPENDIX 4:

JOB CHARACTERISTIC INDEX 1979-1982

JOB CHARACTERISTICS INDEX

Included in the 1979 and 1982 surveys is a measure of perceived job characteristics developed by Sims, Szilagyi, and Keller.¹ This scale, the Job Characteristics Index (JCI), is an extension of the work first begun by Turner and Lawrence in 1965.² The JCI was preceded by an instrument developed by Hackman and Oldham known as the Job Diagnostic Survey (JDS),³ whose dimensions are also incorporated in the JCI, but in a simpler format.

Comparisons of the JCI and JDS by Dunham et al.⁴ have shown that both scales tend to collapse to a one-dimensional scale measuring job-complexity. Therefore, the JCI was shortened by selecting one scale item which loaded strongly on each of the dimensions of job complexity shown to be important in earlier research. In their 1976 article, Sims et al. reported the necessary factor analysis scores used to obtain the abbreviated scale. The seven questionnaire items that comprise the shortened JCI scale are in Section 8, question 23, sub-questions 1-5, and questions 24A and C (Reference #'s R00481.-R00486. and R00488.) for 1979; for 1982, the items are in Section 5, question 36, sub-questions 1-5, and questions 36B and 36D (Reference #'s R07054.-R07059. and R07061.).

NOTES

¹ Sims, Henry R., Szilagyi, Andrew, And Keller, Robert. "The Measurement of Job Characteristics," Academy of Management Journal 26, 2 (June, 1976): 195-212.

² Turner, A.N. and Lawrence, P.R. Industrial Jobs and the Workers: An Investigation of Responses to Task Attributes. Boston: Harvard University Press, 1965.

³ Hackman, J.R. and Oldham, J.R. "Development of the Job Diagnostic Survey," Journal of Applied Psychology 60 (1975): 159-170.

⁴ Dunham, Randall, B., Aldag, Ramon, and Brief, Arthur P. "Dimensionality of Task Design as Measured by the Job Diagnostic Survey," Academy of Management Journal 20, 2 (June, 1977): 209-223.

⁵ Pierce, Jon L. and Dunham, Randall B. "The Measurement of Perceived Job Characteristics: The Job Diagnostic Survey vs. the Job Characteristics Inventory,: Academy of Management Journal 21, 1 (March, 1978): 123-128.

NLSY79 APPENDIX 5:

SUPPLEMENTAL FERTILITY AND

RELATIONSHIP VARIABLES (2000)

FILE CONTENTS

The supplemental fertility data file, previously referred to as FERTILE, has been renamed “Fertility and Relationship History.” This area of interest contains a variety of constructed and edited variables based on the fertility and marriage histories of respondents, as well as the household record, from the 1979-2000 National Longitudinal Surveys of Youth 1979 cohort (NLSY79). These variables enable users to more easily access the wealth of demographic information provided by the surveys and improve the internal consistency of such data across survey years. The file contains dates of birth, sex, and usual living arrangements for all respondents’ children based on a review of the longitudinal data record. Beginning in 1994, the two-digit IDs of the biological children of the female respondents were added to FERTILITY AND RELATIONSHIP HISTORY, as were separate edit flags for male and female respondents. Also included are created variables that summarize dates of marriage(s), number and type of marriage and/or cohabiting relationships, number of live births and other pregnancy outcomes, spacing between births, spacing between first birth and first marriage, and age of the respondent at the time of the first marriage and key fertility events. The variables included in this file are based upon the youth fertility data as revised in a data cleanup program undertaken in 1982-1983 with additional editing provided at selected subsequent survey points.

1982 DATA QUALITY CHECK AND 1986–90 REVISIONS

Many of the inputs into these fertility variables (specifically month and year of child’s birth) were revised in 1982 in order to maximize internal consistency across years. All of the fertility-related variables in the current file are based on these “revised” data items, with the exception of those variables created directly from the respondent’s household record. The variable R08988.01 (Consistency Of Fertility Data 79-82) specifies for each case whether, after revision, any discrepancy in the fertility history from 1979 to 1982 still remains. In general, when a respondent was interviewed each year from 1979 to 1982, the revised 1982 variables give an accurate picture of the respondent’s fertility history as of 1982. However, some inconsistencies in the data over the period were irresolvable. In such cases, the original 1982 data were left intact and a code was assigned to indicate the nature of the remaining inconsistency. A code of “1” means that the 1982 data are consistent with previous survey years, “2” indicates that a dating error remains, and “3” means that an error in the number of children still remains as of 1982. When an error on dating and on number of children occurred simultaneously, the respondent was coded as having an error on number of children since this type of misreport was considered to be more serious.

Further edits to children’s dates of births for female respondents occurred in 1986 as a result of preparation for the first round of the new child-mother data tape, and in 1987 for male respondents. In 1989, attention was given to correcting subsequent inconsistencies for both males and females. With each successive survey round, an effort is made to fill in previously missing values on children’s birth dates and to include children the respondent has previously failed to report. Since 1990, additional information collected on the children of the female NLSY79 respondents and released on the “Children/Young Adults of the NLSY79” file, has been used to help reconcile inconsistent information for those respondents in the Fertility and Relationship History file. In general, the quality of the fertility record for the female respondents is superior to that reported for the male respondents.

1994 DATA QUALITY CHECK

Beginning in 1994, the supplemental fertility file includes several variables not available for earlier years. For each child of a female respondent, there is now a two-digit identification number variable, which will allow users to more accurately link data from the fertility file with data from the NLSY79 Child file. There is also a comprehensive edit flag for female respondents, allowing users to know which female respondents have had changes made to their fertility record for each survey year compared to previous data

releases and what the nature of the change is. Detailed information about the coding categories is provided in the last section of this document. For male respondents, three types of edit flags are provided: two which show the extent of discrepancies between the most recent fertility record available and the current CRF data, and one which indicates cases edited to correct birth order.

As part of the preparation of the 1994 Fertility and Relationship History file, a major data reconciliation was undertaken, comparing the birth records of the female NLSY79 respondents across years. As a result, users may notice discrepancies in these variables across time. It is important for the user to understand that when a date of birth is corrected, we do not change the data for earlier points in time. Thus, there may be inconsistencies in the dates of birth and ages of specific children between the 1994 data and earlier or subsequent reports.

1993 VARIABLE CONSTRUCTION

Although NLSY79 respondents were interviewed in 1993, the Fertility and Relationship History area of interest originally did not contain constructed variables for the 1993 survey year. Data collected in 1993 were used in the 1994 data reconciliation, and some information, such as dates of birth and death, were incorporated into the 1994 or later variables where appropriate. As part of the data work for the NLSY79 2000 data release, however, these 1993 variables were added to the Fertility and Relationship History area of interest.

The set of variables constructed for 1993 is similar to the sets created for other years in which the full battery of pre- and post-natal questions were not asked, such as 1989 or 1991, in that the pregnancy history variables were not created. Dates of birth, gender, and usual residence are constructed for all children, and 2-digit IDs are provided for the children of female respondents. As with the 1994 data, information for the male respondents was not examined as closely as was the information for female respondents. There is a detailed edit flag for the female respondents, as well as the three edit flags for the male respondents. The marriage history variables were also constructed for 1993.

Although the pattern of data evaluation and the creation of edit flags follow that of the 1994 data reconciliation, data comparisons were done only with data from the 1992 survey and earlier. Because of this approach to the data reconciliation and variable construction, the transition from 1993 to 1994 will not be seamless. Users are always advised to use Fertility and Relationship History data from the most recent survey in which a respondent was interviewed.

1979-2000 RELATIONSHIP HISTORY VARIABLES

As an outgrowth of research funded by the National Institute of Child Health and Human Development (NICHD), a series of cross-sectional relationship history variables has been added to the Fertility and Relationship History area of interest. Survey staff carefully examined the names and relationships of household members as reported in the household roster, as well as in the marriage history information collected in various rounds. An attempt was made to identify all cohabiting partners listed in the household record at any point and to combine this information with the marriage data reported by respondents. In this way, the number of spouse/partners reported across survey years was identified.

Two variables per survey year have been constructed for all interviews through 2000. For each survey, the first variable indicates the total number of spouses/partners a respondent has ever been known to have. The second variable reports the respondent's relationship to the current or most recent spouse/partner. The possible relationship codes are spouse (1), partner (33) or other (36); respondents with no known spouses or partners are coded -999. The code of "other" is used for cases where someone is listed in the

household record of a given survey as, for example, an “other non-relative” but is listed as either a partner or a spouse in the preceding or subsequent survey.

These two variables can be used in conjunction to establish a numeric ID for the current or most recent spouse/partner for any given year. The value of the first variable (0-9 as of 2000) is the first digit of the ID, and the value of the second variable (1, 33, or 36) is the remainder. The resulting number indicates the sequential order of the spouse/partner in the respondent’s relationship history and the respondent’s relationship to that person.

Users should note that the total count of spouses and partners may be understated, because these variables are based on information reported on the interview date. A spouse or partner may have appeared between survey rounds but not have been present at any survey point. Early examination suggests that this applies to only a modest proportion of cases. In some instances, identification of spouses who were present only between rounds may be possible by using the NLSY79 marriage history, as well as the marriage transition information available at each survey point.

PUBLISHED REPORTS

Since 1982, the NLSY79 fertility data have been collected with the support of funding from the National Institute of Child Health and Human Development (NICHD). The 1982 data quality check was also completed under the auspices of NICHD. A comprehensive description of the evaluation procedures used in revising the data, as well as a variety of tabular and multivariate analyses, can be found in the reports entitled “Fertility-Related Data in the 1982 National Longitudinal Survey of Work Experience of Youth: An Evaluation of Data Quality and Some Preliminary Analytical Results” and “Evaluation of Fertility Data and Preliminary Analytical Results from the 1983 (5th round) Survey of the National Longitudinal Surveys of Work Experience of Youth,” both prepared by Frank L. Mott, Center for Human Resource Research. The latter report also includes a detailed evaluation of the NLSY79 abortion data. Additional tables referencing the 1986 data can also be found in “Selected Tables: National Longitudinal Surveys of Youth Cohort, May 1987.” Evaluations of the marital history data are provided in R. Jean Haurin, “Inconsistencies in the NLSY Marital History Data - 1986 Supplemental Fertility File” and “Marriage and Childbearing of Adults: An Evaluation of the 1992 National Longitudinal Survey of Youth.” These reports are available from:

NLS User Services
Center for Human Resource Research
921 Chatham Lane, Suite 100
Columbus, Ohio 43221-2418
(614) 442-7366; Fax (614) 442-7329
usersvc@postoffice.chrr.ohio-state.edu

Questions regarding the nature of the fertility data should be directed to Canada Keck, who can be reached in writing at the above address, via e-mail at keck@postoffice.chrr.ohio-state.edu, or by phone at (614) 442-7300. Inquiries relating to the purchase of the data should be addressed to the NLS User Services staff at the above address and phone number.

DATA DESCRIPTION

The current Fertility and Relationship History file of 1266 variables includes a small set of fertility and relationship items for all respondents for 1979-1981 and a more extensive set of variables for 1982-2000. These variables include the marriage, relationship and fertility histories of both male and female respondents and the pregnancy histories of the females. The 2000 Fertility and Relationship History file contains what we believe to be the most accurate information for the female respondents as of that survey point. Information about the biological children of the male respondents was not as closely scrutinized as was the females. Users should note several caveats with regard to the creation of specific variables.

- (1) The fertility data of male respondents has typically been less closely scrutinized than that of the female respondent. Beginning in 1993, separate edit flags for males and females have been added to the Fertility and Relationship History record.
- (2) Users may notice discrepancies in dates of birth or gender variables across time. These discrepancies arise as part of the data reconciliation process. Occasionally, a child who is initially reported as a biological child has been later found to be a stillbirth or a non-biological child and removed from the fertility record. It is important for the user to understand that when a date of birth or other information is corrected, we do not change the data for earlier points in time. Thus, there may be inconsistencies in the dates of birth and ages of specific children, or the total number of children, between the current fertility record and earlier reports.
- (3) Variables indicating the number of children or the age of the youngest child in the household refer to the respondent's biological, adopted, or step-children present in the household at the time of the interview. These variables are created by cycling through the household record for the given survey year. The variable titles and labels were historically adjusted for the 1998 data release to make this distinction more apparent.
- (4) Variables relating to the female pregnancy histories such as number of pregnancies, number of miscarriages/stillbirths, month and year began first pregnancy, age began first pregnancy, and outcome of first pregnancy have valid values only for female respondents interviewed at the survey year in question and who were also interviewed at the time of the 1982 and 1983 surveys when full retrospective pregnancy histories were collected. All male respondents as well as female respondents not interviewed in both 1982 and 1983 are coded as a “-4” on these variables since a complete pregnancy profile is unavailable. Beginning with 1992, miscarriages and stillbirths are collapsed into a single code (“2”) on the variable “Outcome of First Pregnancy.”

Confidential abortion reports were collected in 1984, 1986, 1988, 1990, 1992, 1994, 1996, 1998, and 2000. This information has been incorporated into the creation of the pregnancy-related variables. For that small subset of female respondents for whom full pregnancy histories are unavailable, some will have full abortion data if they were interviewed in 1984. Thus, there are smaller numbers of respondents with a code of “-4” on the variable for number of abortions than on the other pregnancy variables.

Current pregnancies are included in the count of the number of pregnancies as of a given survey date, and twins/triplets represent a single pregnancy. Where questionnaire items for the beginning date of the first pregnancy are unavailable and the outcome was a live birth, 9 months are subtracted from the child's birth date to obtain the beginning date of the first pregnancy. Where the outcome of the first pregnancy was an abortion reported only in one of the confidential reports, 3 months are subtracted.

- (5) Beginning with the release of the 1985 marriage variables, an effort has been made to reconcile marriage dates with the key variables for current marital status made available in the “Key

Variables” area of interest. For approximately 100 cases there are inconsistencies in the marriage histories over time, with some respondents changing their marital status from ever-married to never-married or vice-versa based on the marital status change item provided on the information sheet. Where the change was made very early in the longitudinal record and the respondent continued to verify the changed status in subsequent surveys, the marriage variables for 1985 and subsequent years have been altered accordingly and will differ from the marriage variables provided earlier. Where a change is recorded from never-married to ever-married using the information sheet item only, marriage dates were not collected and thus the respondent is missing information (or coded “-3”) on the date of marriage. For all survey years, a marriage is considered to have ended only if the respondent reports a change to widowhood or divorce.

- (6) All age variables referenced to events are constructed with the original date of birth of the respondent provided at the 1979 survey (R00003., R00004. and R00005). These variables were used to define a respondent’s eligibility to be included in the NLSY79 sample.
- (7) The variables indicating months between first marriage and first birth have traditionally ranged from negative to positive numbers, with specialized codes to indicate non-interview, no first child, no first marriage, and missing date information. Beginning with the 1998 data release, these data indicating months between first marriage and first birth for each FERTILE record 1982–2000 have been converted to all positive numbers. These variables are paired with a flag variable that indicates whether the first marriage occurred before or after the first birth. Both variables have been assigned new reference numbers. The specialized codes for non-interviews have been dropped; however, the other specialized codes have been retained. The original versions of these variables **do not** appear in the public release.
- (8) The newly added relationship history variables help users track the number of spouse/partners identified through our data collection process. However, not all cohabiting partners may have been reported as such, and cohabiting partners could also have entered and exited households between survey rounds and thus be unavailable for identification.

CODEBOOK CATEGORIES

The following variables have special coding specifications that users need to be aware of when using the 1994–2000 supplemental fertility data.

- R50882. Edit Flag for Female Respondents 1994 Survey
- R51735. Edit Flag for Female Respondents 1996 Survey
- R64871. Edit Flag for Female Respondents 1998 Survey
- R70149. Edit Flag for Female Respondents 2000 Survey

This edit flag is a general code that indicates the status of the MOTHER’S fertility record, and the indicated changes do not specify which child was affected.

- 0 = Consistent with previous supplemental fertility file records
- 1 = A child made younger
- 2 = A child made older
- 3 = Corrected previously missing information
- 4 = Information inconsistent with previous supplemental fertility; previous supplemental fertility information used
- 5 = Information inconsistent with previous supplemental fertility; mother’s current information accepted reluctantly
- 6 = IDs assigned out of birth order

- 7 = Discrepancy between CRF and FERTILE; current FERTILE will be consistent with previous supplemental fertility, but CRF preserved for next survey round
8 = Child removed from FERTILE; incorrectly recorded non-biological child
9 = Child removed from FERTILE; incorrectly recorded pregnancy loss
10 = IDs assigned out of birth order AND made a child younger
11 = IDs assigned out of birth order AND made a child older
12 = Gender changed from previous supplemental fertility
13 = Data from mother inconsistent; one child deleted, another added
14 = IDs out of birth order AND discrepancy between FERTILE and CRF (CRF preserved for next survey round)
15 = Child removed from FERTILE; incorrectly recorded pregnancy loss AND changed gender on another child from previous FERTILE
16 = Made current supplemental fertility consistent with current Child Supplement (new child)
17 = Sex missing from current CRF (new child); used information from current CS
18 = Child assessed in current round but inexplicably missing from current CRF; added to current supplemental fertility with residence information from HHR if possible
19 = Hand edited date of death
20 = Day of birth ONLY discrepancy between previous supplemental fertility and current CRF; unedited CRF day used
21 = Child assessed in current survey round but mother is a noninterview
22 = Mother added surprise older child; IDs out of birth order
23 = Incorrect code of 99 (deleted) generated for deceased/adopted out child; information corrected
24 = Duplicate date of birth of existing child on CRF; edited to reflect previous supplemental fertility
25 = Non-biological child not previously on FERTILE deleted from current CRF
26 = Incorrect code of 99 (deleted) generated for live child; status corrected and residence taken from HHR if possible
27 = Incorrect HH flag generated by CAPI for deceased/adopted out child; information corrected (New code in 1996)
28 = Residence missing from CRF; information from fertility section and/or HHR used (New code in 1996)
29 = Partial interview; used previous supplemental fertility file; residence coded from HH record if possible (New code in 1998)
30 = Incorrect code of 8 (deceased) generated for live child; status corrected and residence taken from HHR if possible (new code in 1998)
31 = Information corrected based on YA respondent correcting birth, age not affected
32 = Information corrected based on YA respondent correcting birth, age made older
33 = Information corrected based on YA respondent correcting birth, age made younger
34 = Child removed from FERTILE, added through interviewer error but not caught previously
35 = Residence coded based on information from HHR

There are three edit flags for MALE respondents for 1994–2000. The coding scheme for these flags appears in the codebook.

NLSY79 APPENDIX 6:

URBAN-RURAL AND SMSA-CENTRAL CITY

VARIABLES

1979-2000 URBAN-RURAL VARIABLES

1979-1996 Urban-Rural Residence

Through 1996, the NLSY79 Urban-Rural Residence variables were constructed using the total and urban population data for the county of residence from the 1970 Census of Population Characteristics of the Population (for NLSY79 1979-1982) and from the 1980 Census of Population and Housing (for NLSY79 1983-1988). These data are included in the 1977 and 1983 *County and City Data Book* data files respectively.

The urban population consists of all inhabitants of urbanized areas. An urbanized area is defined as a central core or city and its adjacent, closely settled territory which have a combined total population of 50,000 or more (with exceptions in Alaska, New York, the New England states, and Wisconsin). These definitions have remained largely unchanged since 1950. For more detailed definitions and comments on exceptions, refer to the *U.S. Bureau of the Census 1970/1980 Census of Population, Characteristics of the Population, Number of Inhabitants* (Series PC80-1-A).

Calculation of the 1979–96 NLSY79 Urban-Rural Residence variables involved the following two steps:

- 1) The percent of urban population was calculated by dividing the urban population of the county by the total county population and multiplying by 100.
- 2) Rural counties for the NLSY79 variables were defined as those with between 0-49% urban population. Urban counties were defined as those with 50% or more urban population.

1998-2000 Urban-Rural Residence

Beginning with the 1998 release, the information needed to calculate whether the county was urban/rural was no longer available in the *County and City Data Book* releases. For 1998, a respondent is coded urban if living in an urbanized area or in a place with greater than 2500 population. There is no regard to county.

User Notes: The method of calculating urban vs rural prior to the 1998 data release designated an entire county as either urban or rural. For example, if a county contained multiple respondents, some living in an urban-like area and others in rural areas, they all received the same designation depending on the degree of urbanization of the county of residence. In 1998 and 2000, the method of calculating urban/rural allows a respondent living in a rural area of a county to be coded rural, while another respondent in an urban area of that same county can be coded urban. **The net affect appears to be an increase in respondents living in rural areas.**

For all survey years, non-interview cases are set to a -5 value on the Urban-Rural Residence variable. All other missing cases are valid skips on these variables.

For a detailed discussion of the procedures used for hand-editing and merging respondent data with other data in creating the geographic variables, see “Appendix 10: Geocode Documentation” in the NLSY79 *Geocode Codebook Supplement*.

1979-2000 SMSA-CENTRAL CITY VARIABLES

The NLSY79 SMSA-Central City variables are constructed using data for the SMSA/MSA and Place Description (PD) included in the City Reference File (CRF) data files. The 1973 CRF was used for NLSY79 1979-1982 SMSA-Central City variables; the 1982 CRF was used for 1983 variables; 1983 CRF was used for 1984-1987 variables; the 1987 CRF was used for the 1988-1992 variables; and the 1992 CRF was used for the 1993-1998 variables. See below for information on the 2000 variables. (For a more detailed discussion of changes in official terminology and geographic designations and comparability across years, see “Appendix 10: Geocode Documentation” in the *NLSY79 Geocode Codebook Supplement*.)

1979-1998 SMSA-Central City Residence

Through 1998, calculation of the SMSA-Central City Residence variables involved the following two steps:

- 1) A respondent’s SMSA/MSA and PD were assigned based on the county, state, and zip code of current residence. (For a detailed discussion of the procedures used for hand-editing and merging respondent data with other data in creating the geographic variables, including the SMSA/MSA, see “Appendix 10: Geocode Documentation” in the *NLSY79 Geocode Codebook Supplement*.)
- 2) Based on their PD and SMSA/MSAs, respondents were assigned to one of the following categories:
 - Respondents not residing in an SMSA/MSA were defined as “not in SMSA.”
 - Respondents residing in an SMSA/MSA, but not in a central city of an SMSA/MSA according to the PD, were defined as “SMSA, not central city.”
 - Respondents for whom the PD leads to an ambiguous central city residence status were defined as “SMSA, central city not known.” These cases generally resulted from zip codes that cover more than one geographically-defined area.
 - Respondents residing in both an SMSA/MSA and the central city of an SMSA/MSA according to the PD were defined as “SMSA in central city.”

Non-interview respondents were assigned a value of -5. Respondents with an ambiguous SMSA/MSA residence status or missing values for reasons other than non-interview are valid skips on these variables.

2000 SMSA-Central City Residence

Beginning with the 2000 release, the calculation of the central city variable was revised slightly. The process still consists of two components. The first delineation is MSA/non-MSA residence. For respondents in an MSA, the residence is further defined by placement inside or outside a central city as defined by the Census Bureau.

Respondents who live in an MSA who have a quality of match code of either manual edit or zip centroid are evaluated before assigning the central city code. If the street (manual edit match) or the zip code area (zip centroid match) falls entirely within the boundaries of the central city they are coded as central city. If the street or zip code area fall completely outside the central city boundary, they are coded as not in a central city. Otherwise, these respondents are coded as central city unknown.

A further complication is that Maptitude, the software currently being used, does not include central city boundary files for several areas. Regardless of quality of match code, respondents in those areas are coded as being in a MSA with an unknown central city status since the mailing address is insufficient to reveal whether the respondent resides within the central city boundaries. When updates of Maptitude are available, the city boundaries will be updated.

NLSY79 APPENDIX 7:

UNEMPLOYMENT RATE

UNEMPLOYMENT RATE

The NLSY79 unemployment rate variables are constructed using state and area labor force data from the May publication of *Employment and Earnings* for the month of March of each survey year. *Employment and Earnings* is published by the U.S. Department of Labor, Bureau of Labor Statistics and lists the civilian labor force and number of unemployed persons for every state and selected metropolitan areas.

The figures provided for the state and selected metropolitan areas are used to compute the unemployment rate for the portion or balance of the state that is not represented in a metropolitan statistical area. If a respondent resides in a metropolitan area that is listed in *Employment and Earnings*, then the unemployment rate is the unemployment rate for that metropolitan area. Otherwise, the unemployment rate is the computed balance of state unemployment rate for the state in which the respondent resides.

The respondent's metropolitan statistical area is assigned based on the county, state, and zip code of current residence. (For a detailed discussion of the hand-editing and merging process with other data files used to create the geographic variables, including the metropolitan statistical area, see Appendix 10: Geocode Documentation.)

As with other geographic based variables, respondents who are in the military, who are living outside of the United States, or who have invalid geographic data for a given survey year are valid skips on these variables.

The geocode data files contain the continuous unemployment rate as it was calculated for the area of residence. For confidentiality reasons, the main public use data files contain a collapsed version of the continuous unemployment rate.

NLSY79 APPENDIX 8:
HIGHEST GRADE COMPLETED &
ENROLLMENT STATUS VARIABLE
CREATION: 1990-2000

The series of programming statements contained in this appendix were used in the standard computations for the created Highest Grade Completed (HGC) and Enrollment Status as of May 1st Survey Year variables from 1990–2000.

In addition to the standard created variables, revised versions of the HGC and Enrollment Status variables for all survey years (1979–2000) have been added to the NLSY79 main data file. The general sources of error in the standard created variables were:

- ◆ grade “reversals”, in which a respondent completed a lower grade in a later year, rather than staying in the same grade or advancing: The programs included below did not account for these cases, the largest source of which have been respondents enrolled in college
- ◆ respondents with incomplete or ambiguous school information, preventing the computation in a given year of a HGC or Enrollment Status variable (a problem which can then continue through subsequent years, even if the respondent reported attending school in later survey years)

Decision rules for adjustments reflected in the revised variables include:

- ◆ Cases with grade reversals were coded as the highest grade completed previously reported by the respondent
- ◆ A minimum HGC code of 12 was assigned to cases in which no high school diploma or GED had been received, but some college attendance was reported
- ◆ An HGC code of 16 was assigned to cases in which a four-year degree had obviously been earned in 5 or more years
- ◆ Cases in which the highest grade completed was reported as “ungraded” were assigned the previous highest grade completed reported by the respondent
- ◆ HGC values were evaluated in comparison to the May 1st of survey year date and corrected if necessary
- ◆ Cases in which the longitudinal record is highly erratic and HGC could not be computed or revised reliably were assigned a code of “-3” (invalid missing)

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1990

```
/* The following spss code is applied to the data before the pli program is executed. The purpose of the code */  
/* is to appropriately recode selected 1979 and 1980 variables. */  
/* do if (R2283 eq 0 or R2282 eq 0)      compute DLEMO79=R2287    compute DLEYR79=R2288 */  
/* else                                compute DLEMO79=R169     compute DLEYR79=R170 */  
/* end if */  
/* do if (R4052 gt 0 and R2283 ne 0 and R2282 ne 0)  compute DLEMO80=R2287 */  
/*                                         compute DLEYR80=R2288 */  
/* end if */  
/* do if (R2275 eq 0 or R2276 eq 0)      compute GRADE79=R2277 */  
/* else                                compute GRADE79=R2286 */  
/* end if */  
/* do if (R2280 ge 0)                  compute HGCS79=R2280 */  
/* else                                compute HGCS79=R173 */  
/* end if */  
/* if(R2276 eq 0 and R2286 gt 0) R156=1 */  
*****
```

DCL 1 INREC7989,

```
2 CURAT(79:89) PIC '-----9', /* R( 156.) R( 2285.) R( 4168.) R( 6639.) R( 9053.) R(12052.) */  
/* R(16045.) R(19050.) R(23059.) R(25084.) R(29075.) */  
2 DIP(79:89)  PIC '-----9', /* R( 183.) R( 2300.) R( 4182.) R( 6653.) R( 9067.) R(12066.) */  
/* R(16059.) R(19061.) R(23070.) R(25095.) R(29086.) */  
2 DLEMO(79:89) PIC '-----9', /* DLEMO79 DLEMO80 R( 4170.) R( 6641.) R( 9055.) R(12054.) */  
/* R(16047.) R(19052.) R(23061.) R(25086.) R(29077.) */  
2 DLEYR(79:89) PIC '-----9', /* DLEYR79 DLEYR80 R( 4171.) R( 6642.) R( 90546) R(12055.) */  
/* R(16048.) R(19053.) R(23062.) R(25087.) R(29078.) */  
2 GRADE(79:89) PIC '-----9', /* GRADE79 R( 2286.) R( 4169.) R( 6640.) R( 9054.) R(12053.) */  
/* R(16046.) R(19051.) R(23060.) R(25085.) R(29076.) */  
2 HGA(79:89)  PIC '-----9', /* R( 172.) R( 2291.) R( 4173.) R( 6644.) R( 9058.) R(12057.) */  
/* R(16050.) R(19055.) R(23064.) R(25089.) R(29080.) */  
2 HGC(79:89)  PIC '-----9', /* R( 2167.) R( 4064.) R( 6189.) R( 8982.) R(11450.) R(15202.) */  
/* R(18909.) R(22580.) R(24454.) R(28711.) R(30748.) */  
2 HGCS(79:89) PIC '-----9', /* HGCS79 R( 2292.) R( 4174.) R( 6645.) R( 9059.) R(12058.) */  
/* R(16051.) R(19056.) R(23065.) R(25090.) R(29081.) */  
2 INTMO(79:89) PIC '-----9', /* R( 1725.) R( 3292.) R( 5307.) R( 8099.) R(10457.) R(14275.) */  
/* R(17946.) R(21562.) R(23657.) R(27425.) R(29861.) */  
2 WEIGHT(79:89) PIC '9999999'; /* R( 2161.) R( 4052.) R( 6146.) R( 8967.) R(11444.) R(15196.) */  
/* R(18902.) R(22573.) R(24445.) R(28700.) R(30738.) */
```

DCL 1 INREC90,

```
2 WEIGHT_90 PIC '9999999', /* R(34002.) */  
2 SCHOOL_90,  
 5 ASLI_90  PIC '-----9', /* R(31070.) */      5 CURAT_90  PIC '-----9', /* R(31096.) */  
 5 DIP_90   PIC '-----9', /* R(31107.) */      5 DLEMO_90  PIC '-----9', /* R(31098.) */  
 5 DLEYR_90 PIC '-----9', /* R(31099.) */      5 GRADE_90  PIC '-----9', /* R(31097.) */  
 5 HGA_90   PIC '-----9', /* R(31101.) */      5 HGCS_90   PIC '-----9', /* R(31102.) */  
 5 INTMO_90 PIC '-----9', /* R(33025.) */      5 HAVEDIP_90 PIC '-----9', /* R(31106.) */  
 5 ATTCOL_90 PIC '-----9'; /*compute ATTCOL_91=0*/ /*if R(31103.)>3 then ATTCOL_91=1*/  
 5 GRA_90   PIC '-----9', /* compute GRA_91=-3 */  
/* if R(31096.)=1 then GRA_91=R(31097.) */  
/* else if R(31096.)=0 then GRA_91=R(31101.) */  
/* else if R(31070.)=0 then GRA_91=-4 */
```

```
/* CREATION OF HIGHEST GRADE COMPLETED and ENROLLMENT STATUS AS OF 5-1-90 */  
HGC_90=-4; ENROLL_90=-4;
```

```

do I=80 to 89;
  if WEIGHT(I)=0 then do; HGCS(I)=-5; GRADE(I)=-5; end;
end;
if WEIGHT_90=0 then do; HGCS_90=-5; GRADE_90=-5; ENROLL_90=-5; HGC_90=-5; end;

if WEIGHT_90>0 then do;
  if ASLI_90=0 & (HAVEDIP_90=-4 ! HAVEDIP_90=0) then do;
    if WEIGHT(89)>0 then do;
      if INTMO(89)>=5 & (CURAT(89)=1 ! (DLEMO(89)>=5 & DLEYR(89)=89)) &
          HGCS(89)>HGC(89) then HGC_90=HGCS(89);           else HGC_90=HGC(89);
    end;
    else if WEIGHT(88)>0 then do;
      if INTMO(88)>=5 & (CURAT(88)=1 ! (DLEMO(88)>=5 & DLEYR(88)=88)) &
          HGCS(88)>HGC(88) then HGC_90=HGCS(88);           else HGC_90=HGC(88);
    end;
    else if WEIGHT(87)>0 then do;
      if INTMO(87)>=5 & (CURAT(87)=1 ! (DLEMO(87)>=5 & DLEYR(87)=87)) &
          HGCS(87)>HGC(87) then HGC_90=HGCS(87);           else HGC_90=HGC(87);
    end;
    else if WEIGHT(86)>0 then do;
      if INTMO(86)>=5 & (CURAT(86)=1 ! (DLEMO(86)>=5 & DLEYR(86)=86)) &
          HGCS(86)>HGC(86) then HGC_90=HGCS(86);           else HGC_90=HGC(86);
    end;
    else if WEIGHT(85)>0 then do;
      if INTMO(85)>=5 & (CURAT(85)=1 ! (DLEMO(85)>=5 & DLEYR(85)=85)) &
          HGCS(85)>HGC(85) then HGC_90=HGCS(85);           else HGC_90=HGC(85);
    end;
    else if WEIGHT(84)>0 then do;
      if INTMO(84)>=5 & (CURAT(84)=1 ! (DLEMO(84)>=5 & DLEYR(84)=84)) &
          HGCS(84)>HGC(84) then HGC_90=HGCS(84);           else HGC_90=HGC(84);
    end;
    else if WEIGHT(83)>0 then do;
      if INTMO(83)>=5 & (CURAT(83)=1 ! (DLEMO(83)>=5 & DLEYR(83)=83)) &
          HGCS(83)>HGC(83) then HGC_90=HGCS(83);           else HGC_90=HGC(83);
    end;
    else if WEIGHT(82)>0 then do;
      if INTMO(82)>=5 & (CURAT(82)=1 ! (DLEMO(82)>=5 & DLEYR(82)=82)) &
          HGCS(82)>HGC(82) then HGC_90=HGCS(82);           else HGC_90=HGC(82);
    end;
    else if WEIGHT(81)>0 then do;
      if INTMO(81)>=5 & (CURAT(81)=1 ! (DLEMO(81)>=5 & DLEYR(81)=81)) &
          HGCS(81)>HGC(81) then HGC_90=HGCS(81);           else HGC_90=HGC(81);
    end;
    else if WEIGHT(80)>0 then do;
      if INTMO(80)>=5 & (CURAT(80)=1 ! (DLEMO(80)>=5 & DLEYR(80)=80)) &
          HGCS(80)>HGC(80) then HGC_90=HGCS(80);           else HGC_90=HGC(80);
    end;
    else do;
      if INTMO(79)>=5 & (CURAT(79)=1 ! (DLEMO(79)>=5 & DLEYR(79)=79)) &
          HGCS(79)>HGC(79) then HGC_90=HGCS(79);           else HGC_90=HGC(79);
    end;
    if HGC_90>=12 then ENROLL_90=4;           else if HGC_90>=0 then ENROLL_90=1;
  end;
  else if ASLI_90=1 then do;
    if HGCS_90<=0 then do;
      if CURAT_90=1 & GRADE_90>=1 then do; HGC_90=GRADE_90-1;
      if HGC_90>=12 then ENROLL_90=3;           else ENROLL_90=2;
    end;
  end;
end;

```

```

end;
else if CURAT_90=0 & HGA_90>=1 then do;
    if DLEYR_90<90 ! DLEMO_90<5 then do; HGC_90=HGA_90;
        if HGC_90>=12 then ENROLL_90=4; else ENROLL_90=1;
    end;
    else do; HGC_90=HGA_90-1;
        if HGC_90>=12 then ENROLL_90=4; else ENROLL_90=1;
    end;
end;
else do;
    if HGCS_90=HGA_90 ! HGCS_90=GRADE_90 then do;
        if (DLEYR_90=90 & DLEMO_90>=5) ! (INTMO_90>=5 & CURAT_90=1) then do;
            HGC_90=HGCS_90-1;
            if HGC_90>=12 then ENROLL_90=3; else ENROLL_90=2;
        end;
        else HGC_90=HGCS_90;
    end;
    if HGCS_90=HGA_90-1 ! HGCS_90=GRADE_90-1 then HGC_90=HGCS_90;
    else if (HGCS_90<HGA_90 & HGA_90>0 & HGA_90<20) ! (HGCS_90<GRADE_90 & GRADE_90>0 & GRADE_90<20) then HGC_90=HGCS_90;
    if ENROLL_90=-4 & CURAT_90=1 & HGC_90>=0 then do;
        if HGC_90>=12 then ENROLL_90=3; else ENROLL_90=2;
    end;
    if ENROLL_90=-4 & CURAT_90=0 & HGC_90>=0 then do;
        if HGC_90>=12 then ENROLL_90=4; else ENROLL_90=1;
    end;
end;
end;
if ((HGC_90=10 ! HGC_90=11) & (HAVEDIP_90=1 ! ATTCOL_90=1)) ! (DIP_90=2 & HGC_90<12) !
    (HGC_90=-4 & (DIP_90=1 ! DIP_90=3)) then do;
    HGC_90=12;
    if ENROLL_90>0 then ENROLL_90=5-ENROLL_90;
    else if CURAT_90=1 ! (DLEYR_90=90 & DLEMO_90>=5) then ENROLL_90=3; else ENROLL_90=4;
end;
else if HGC_90<0 & DIP_90=-3 then do; HGC_90=-3; ENROLL_90=-3; end;
if HGC_90>20 then do; HGC_90=-3; ENROLL_90=-3; end;
if HGC_90>HGC(89)+3 & HGC(89)>0 & DIP_90^=2 & DIP_90^=3 then do;
    HGC_90=-3; ENROLL_90=-3;
end;
if HGC_90>0 & HGC_90<HGC(89)-1 ! (HGC(89)=12 & HGC_90=11) then do;
    HGC_90=-3; ENROLL_90=-3;
end;
if (DIP(79)>1 ! DIP(80)>1 ! DIP(81)>1 ! DIP(82)>1 ! DIP(83)>1 ! DIP(84)>1 ! DIP(85)>1 ! DIP(86)>1 !
    DIP(87)>1 ! DIP(88)>1 ! DIP(89)>1) & HGC_90<12 then do;
    HGC_90 =12; if ENROLL_90=1 ! ENROLL_90<0 then ENROLL_90=4;
end;
if HGC_90=-3 & ENROLL_90=-4 then ENROLL_90=-3;
else if HGC_90=-4 & ENROLL_90=-4 then do; HGC_90=-3; ENROLL_90=-3; end;
end;

HGC_90=R(34015.); ENROLL_90=R(34016.);

/*
Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected school variables. Values of -3 are recoded to valid values where auxilliary information indicates. However, there are a number of -3's computed for highest grade completed and enrollment status that remain.
*/

```

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1991

```
/* The following spss code is applied to the data before the PLI program is executed. The purpose of the code */  
/* is to appropriately recode selected 1979 and 1980 variables. */  
/* do if (R2283 eq 0 or R2282 eq 0)      compute DLEMO79=R2287    compute DLEYR79=R2288 */  
/* else                                compute DLEMO79=R169      compute DLEYR79=R170 */  
/* end if */  
/* do if (R4052 gt 0 and R2283 ne 0 and R2282 ne 0) */  
/* compute DLEMO80=R2287      compute DLEYR80=R2288      end if */  
/* do if (R2275 eq 0 or R2276 eq 0)      compute GRADE79=R2277 */  
/* else                                compute GRADE79=R2286      end if */  
/* do if (R2280 ge 0)                  compute HGCS79=R2280 */  
/* else                                compute HGCS79=R173      end if */  
/* if (R2276 eq 0 and R2286 gt 0) R156=1 */  
*****
```

DCL 1 INREC7990,

```
2 CURAT(79:90) PIC '-----9', /*R( 156.) R( 2285.) R( 4168.) R( 6639.) R( 9053.) R(12052.) */  
/*R(16045.) R(19050.) R(23059.) R(25084.) R(29075.) R(31096.) */  
2 DIP(79:90) PIC '-----9', /*R( 183.) R( 2300.) R( 4182.) R( 6653.) R( 9067.) R(12066.) */  
/*R(16059.) R(19061.) R(23070.) R(25095.) R(29086.) R(31107.) */  
2 DLEMO(79:90) PIC '-----9', /*DLEMO79 DLEMO80 R( 4170.) R( 6641.) R( 9055.) R(12054.) */  
/*R(16047.) R(19052.) R(23061.) R(25086.) R(29077.) R(31098.) */  
2 DLEYR(79:90) PIC '-----9', /*DLEYR79 DLEYR80 R( 4171.) R( 6642.) R( 90546) R(12055.) */  
/*R(16048.) R(19053.) R(23062.) R(25087.) R(29078.) R(31099.) */  
2 GRADE(79:90) PIC '-----9', /*GRADE79 R( 2286.) R( 4169.) R( 6640.) R( 9054.) R(12053.) */  
/*R(16046.) R(19051.) R(23060.) R(25085.) R(29076.) R(31097.) */  
2 HGA(79:90) PIC '-----9', /*R( 172.) R( 2291.) R( 4173.) R( 6644.) R( 9058.) R(12057.) */  
/*R(16050.) R(19055.) R(23064.) R(25089.) R(29080.) R(31101.) */  
2 HGC(79:90) PIC '-----9', /*R( 2167.) R( 4064.) R( 6189.) R( 8982.) R(11450.) R(15202.) */  
/*R(18909.) R(22580.) R(24454.) R(28711.) R(30748.) R(34015.) */  
2 HGCS(79:90) PIC '-----9', /*HGCS79 R( 2292.) R( 4174.) R( 6645.) R( 9059.) R(12058.) */  
/*R(16051.) R(19056.) R(23065.) R(25090.) R(29081.) R(31102.) */  
2 INTMO(79:90) PIC '-----9', /*R( 1725.) R( 3292.) R( 5307.) R( 8099.) R(10457.) R(14275.) */  
/*R(17946.) R(21562.) R(23657.) R(27425.) R(29861.) R(33025.) */  
2 WEIGHT(79:90) PIC '9999999'; /*R( 2161.) R( 4052.) R( 6146.) R( 8967.) R(11444.) R(15196.) */  
/*R(18902.) R(22573.) R(24445.) R(28700.) R(30738.) R(34002.) */
```

DCL 1 INREC91,

```
2 WEIGHT_91 PIC '9999999', /* R(36558.) */  
2 SCHOOL_91,  
5 ASLI_91 PIC '-----9', /*R(35070.)*/ 5 CURAT_91 PIC '-----9', /*R(35096.)*/  
5 DIP_91 PIC '-----9', /*R(35107.)*/ 5 DLEMO_91 PIC '-----9', /*R(35098.)*/  
5 DLEYR_91 PIC '-----9', /*R(35099.)*/ 5 GRADE_91 PIC '-----9', /*R(35097.)*/  
5 HGA_91 PIC '-----9', /*R(35101.)*/ 5 HGCS_91 PIC '-----9', /*R(35102.)*/  
5 INTMO_91 PIC '-----9', /*R(35734.)*/ 5 HAVEDIP_91 PIC '-----9', /*R(35106.)*/  
5 ATTCOL_91 PIC '-----9'; /* compute ATTCOL_91=0 */  
/* if R(35103.)>3 then ATTCOL_91=1 */  
5 GRA_91 PIC '-----9', /* compute GRA_91=-3 */  
/* if R(35096.)=1 then GRA_91=R(35097.) */  
/* else if R(35096.)=0 then GRA_91=R(35101.) */  
/* else if R(35070.)=0 then GRA_91=-4 */
```

```
/* CREATION OF HIGHEST GRADE COMPLETED and ENROLLMENT STATUS AS OF 5-1-91 */  
HGC_91=-4; ENROLL_91=-4;  
do I=80 to 90; if WEIGHT(I)=0 then do; HGCS(I)=-5; GRADE(I)=-5; end; end;  
if WEIGHT_91=0 then do; HGCS_91=-5; GRADE_91=-5; ENROLL_91=-5; HGC_91=-5; end;
```

```

if WEIGHT_91>0 then do;
  if ASLI_91=0 & (HAVEDIP_91=-4 ! HAVEDIP_91=0) then do;
    if WEIGHT(90)>0 then do;
      if INTMO(90)>=5 & (CURAT(90)=1 ! (DLEMO(90)>=5 & DLEYR(90)=90)) &
        HGCS(90)>HGC(90) then HGC_91=HGCS(90);                                else HGC_91=HGC(90);
    end;
  if WEIGHT(89)>0 then do;
    if INTMO(89)>=5 & (CURAT(89)=1 ! (DLEMO(89)>=5 & DLEYR(89)=89)) &
      HGCS(89)>HGC(89) then HGC_91=HGCS(89);                                else HGC_91=HGC(89);
  end;
  else if WEIGHT(88)>0 then do;
    if INTMO(88)>=5 & (CURAT(88)=1 ! (DLEMO(88)>=5 & DLEYR(88)=88)) &
      HGCS(88)>HGC(88) then HGC_91=HGCS(88);                                else HGC_91=HGC(88);
  end;
  else if WEIGHT(87)>0 then do;
    if INTMO(87)>=5 & (CURAT(87)=1 ! (DLEMO(87)>=5 & DLEYR(87)=87)) &
      HGCS(87)>HGC(87) then HGC_91=HGCS(87);                                else HGC_91=HGC(87);
  end;
  else if WEIGHT(86)>0 then do;
    if INTMO(86)>=5 & (CURAT(86)=1 ! (DLEMO(86)>=5 & DLEYR(86)=86)) &
      HGCS(86)>HGC(86) then HGC_91=HGCS(86);                                else HGC_91=HGC(86);
  end;
  else if WEIGHT(85)>0 then do;
    if INTMO(85)>=5 & (CURAT(85)=1 ! (DLEMO(85)>=5 & DLEYR(85)=85)) &
      HGCS(85)>HGC(85) then HGC_91=HGCS(85);                                else HGC_91=HGC(85);
  end;
  else if WEIGHT(84)>0 then do;
    if INTMO(84)>=5 & (CURAT(84)=1 ! (DLEMO(84)>=5 & DLEYR(84)=84)) &
      HGCS(84)>HGC(84) then HGC_91=HGCS(84);                                else HGC_91=HGC(84);
  end;
  else if WEIGHT(83)>0 then do;
    if INTMO(83)>=5 & (CURAT(83)=1 ! (DLEMO(83)>=5 & DLEYR(83)=83)) &
      HGCS(83)>HGC(83) then HGC_91=HGCS(83);                                else HGC_91=HGC(83);
  end;
  else if WEIGHT(82)>0 then do;
    if INTMO(82)>=5 & (CURAT(82)=1 ! (DLEMO(82)>=5 & DLEYR(82)=82)) &
      HGCS(82)>HGC(82) then HGC_91=HGCS(82);                                else HGC_91=HGC(82);
  end;
  else if WEIGHT(81)>0 then do;
    if INTMO(81)>=5 & (CURAT(81)=1 ! (DLEMO(81)>=5 & DLEYR(81)=81)) &
      HGCS(81)>HGC(81) then HGC_91=HGCS(81);                                else HGC_91=HGC(81);
  end;
  else if WEIGHT(80)>0 then do;
    if INTMO(80)>=5 & (CURAT(80)=1 ! (DLEMO(80)>=5 & DLEYR(80)=80)) &
      HGCS(80)>HGC(80) then HGC_91=HGCS(80);                                else HGC_91=HGC(80);
  end;
  else do;
    if INTMO(79)>=5 & (CURAT(79)=1 ! (DLEMO(79)>=5 & DLEYR(79)=79)) &
      HGCS(79)>HGC(79) then HGC_91=HGCS(79);                                else HGC_91=HGC(79);
  end;
  if HGC_91>=12 then ENROLL_91=4;                                         else if HGC_91>=0 then ENROLL_90=1;
end;
else if ASLI_91=1 then do;
  if HGCS_91<=0 then do;
    if CURAT_91=1 & GRADE_91>=1 then do;
      if HGC_91>=12 then ENROLL_91=3;                                     HGCS_91=GRADE_91-1;
      else ENROLL_91=2;
    end;
  end;

```

```

else if CURAT_91=0 & HGA_91>=1 then do;
    if DLEYR_91<90 ! DLEMO_91<5 then do;
        if HGC_91>=12 then ENROLL_91=4;
    end;
    else do;
        if HGC_91>=12 then ENROLL_91=4;
    end;
end;
else do;
    if HGCS_91=HGA_91 ! HGCS_91=GRADE_91 then do;
        if (DLEYR_91=91 & DLEMO_91>=5) ! (INTMO_91>=5 & CURAT_91=1) then do;
            HGC_91=HGCS_91-1;
            if HGC_91>=12 then ENROLL_91=3;           else ENROLL_91=2;
        end;
        else HGC_91=HGCS_91;
    end;
    if HGCS_91=HGA_91-1 ! HGCS_91=GRADE_91-1 then HGC_91=HGCS_91;
    else if (HGCS_91<HGA_91 & HGA_91>0 & HGA_91<20) ! (HGCS_91<GRADE_91 & GRADE_91>0 &
        GRADE_91<20) then HGC_91=HGCS_91;
    if ENROLL_91=-4 & CURAT_91=1 & HGC_91>=0 then do;
        if HGC_91>=12 then ENROLL_91=3;           else ENROLL_91=2;
    end;
    if ENROLL_91=-4 & CURAT_91=0 & HGC_91>=0 then do;
        if HGC_91>=12 then ENROLL_91=4;           else ENROLL_91=1;
    end;
end;
end;
if ((HGC_91=10 ! HGC_91=11) & (HAVEDIP_91=1 ! ATTCOL_91=1)) ! (DIP_91=2 & HGC_91<12) !
    (HGC_91=-4 & (DIP_91=1 ! DIP_91=3)) then do; HGC_91=12;
    if ENROLL_91>0 then ENROLL_91=5-ENROLL_90;
    else if CURAT_91=1 ! (DLEYR_91=91 & DLEMO_91>=5) then ENROLL_91=3;
    else ENROLL_91=4;
end;
else if HGC_91<0 & DIP_91=-3 then do; HGC_91=-3; ENROLL_91=-3; end;
if HGC_91>20 then do; HGC_91=-3; ENROLL_91=-3; end;
if HGC_91>HGC(90)+3 & HGC(90)>0 & DIP_91^=2 & DIP_91^=3 then do;
    HGC_91=-3; ENROLL_91=-3;
end;
if HGC_91>0 & HGC_91<HGC(90)-1 ! (HGC(90)=12 & HGC_91=11) then do;
    HGC_91=-3; ENROLL_91=-3;
end;
if (DIP(79)>1 ! DIP(80)>1 ! DIP(81)>1 ! DIP(82)>1 ! DIP(83)>1 ! DIP(84)>1 ! DIP(85)>1 ! DIP(86)>1 !
    DIP(87)>1 ! DIP(88)>1 ! DIP(89)>1 ! DIP(90)>1) & HGC_91<12 then do; HGC_91=12;
    if ENROLL_91=1 ! ENROLL_91<0 then ENROLL_91=4;
end;
if HGC_91=-3 & ENROLL_91=-4 then ENROLL_91=-3;
else if HGC_91=-4 & ENROLL_91=-4 then do; HGC_91=-3; ENROLL_91=-3; end;
end;

HGC_91=R(36569.) ENROLL_91=R(36570.)

```

/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected school */
 /* variables. Values of -3 are recoded to valid values where auxilliary information indicates. However, there */
 /* are a number of -3's computed for highest grade completed and enrollment status that remain. */

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1992

```
/* The following code is applied to the data before the PLI program is executed. The purpose of the code is to */  
/* appropriately recode selected 1979 and 1980 variables. */  
/* if (R2283 eq 0 or R2282 eq 0) then do; DLEMO79=R2287; DLEYR79=R2288; end; */  
/* else do; DLEMO79=R169; DLEYR79=R170; end; */  
/* if (R4052 gt 0 and R2283 ne 0 and R2282 ne 0) then do; DLEMO80=R2287; DLEYR80=R2288; end; */  
/* if (R2275 eq 0 or R2276 eq 0) then GRADE79=R2277 ; else GRADE79=R2286; */  
/* if (R2280 ge 0) then HGCS79=R2280; else compute HGCS79=R173; end if; */  
/* if (R2276 eq 0 and R2286 gt 0) R156=1 */  
*****
```

DCL 1 INREC79OLD,

```
2 CURAT(79:LASTYR) PIC '-----9', /* R( 156.) R( 2285.) R( 4168.) R( 6639.) R( 9053.) R(12052.)  
R(16045.) R(19050.) R(23059.) R(25084.) R(29075.) R(31096.) R(35096.) */  
2 DIP(79:LASTYR) PIC '-----9', /* R( 183.) R( 2300.) R( 4182.) R( 6653.) R( 9067.) R(12066.)  
R(16059.) R(19061.) R(23070.) R(25095.) R(29086.) R(31107.) R(35107.) */  
2 DLEMO(79:LASTYR) PIC '-----9', /* DLEMO79 DLEMO80 R( 4170.) R( 6641.) R( 9055.) R(12054.)  
R(16047.) R(19052.) R(23061.) R(25086.) R(29077.) R(31098.) R(35098.) */  
2 DLEYR(79:LASTYR) PIC '-----9', /* DLEYR79 DLEYR80 R( 4171.) R( 6642.) R( 90546) R(12055.)  
R(16048.) R(19053.) R(23062.) R(25087.) R(29078.) R(31099.) R(35099.) */  
2 GRADE(79:LASTYR) PIC '-----9', /* GRADE79 R( 2286.) R( 4169.) R( 6640.) R( 9054.) R(12053.)  
R(16046.) R(19051.) R(23060.) R(25085.) R(29076.) R(31097.) R(35097.) */  
2 HGA(79:LASTYR) PIC '-----9', /* R( 172.) R( 2291.) R( 4173.) R( 6644.) R( 9058.) R(12057.)  
R(16050.) R(19055.) R(23064.) R(25089.) R(29080.) R(31101.) R(35101.) */  
2 HGC(79:LASTYR) PIC '-----9', /* R( 2167.) R( 4064.) R( 6189.) R( 8982.) R(11450.) R(15202.)  
R(18909.) R(22580.) R(24454.) R(28711.) R(30748.) R(34015.) R(36569.) */  
2 HGCS(79:LASTYR) PIC '-----9', /* HGCS79 R( 2292.) R( 4174.) R( 6645.) R( 9059.) R(12058.)  
R(16051.) R(19056.) R(23065.) R(25090.) R(29081.) R(31102.) R(35102.) */  
2 INTMO(79:LASTYR) PIC '-----9', /* R( 1725.) R( 3292.) R( 5307.) R( 8099.) R(10457.) R(14275.)  
R(17946.) R(21562.) R(23657.) R(27425.) R(29861.) R(33025.) R(35734.) */  
2 WEIGHT(79:LASTYR) PIC '9999999', /* R( 2161.) R( 4052.) R( 6146.) R( 8967.) R(11444.) R(15196.) */  
R(18902.) R(22573.) R(24445.) R(28700.) R(30738.) R(34002.) R(36558.) */  
2 WEIGHT92 PIC '9999999'; /* R(36558.) */
```

DCL 1 INRECNNEW ,

```
2 NORCID_CUR PIC '-----9',  
2 INTOB_CUR PIC '9999999', /* DUMMY */  
2 SCHOOL_CUR,  
5 ASLI_CUR PIC '-----9', /* R(37070.) */  
5 CURAT_CUR PIC '-----9', /* R(37096.) */  
5 DIP_CUR PIC '-----9', /* R(37107.) */  
5 DLEMO_CUR PIC '-----9', /* R(37098.) */  
5 DLEYR_CUR PIC '-----9', /* R(37099.) */  
5 GRA_CUR PIC '-----9', /* compute GRA_CUR=-3 */  
/* if R(37096.)=1 then GRA_CUR=R(37096.) */  
/* else if R(37096.)=0 then GRA_CUR=R(37100.) */  
/* else if R(37070.)=0 then GRA_CUR=-4 */  
5 GRADE_CUR PIC '-----9', /* R(37097.) */  
5 HGA_CUR PIC '-----9', /* R(37101.) */  
5 HGCS_CUR PIC '-----9', /* R(37102.) */  
5 INTMO_CUR PIC '-----9', /* R(39176.) */  
5 HAVEDIP_CUR PIC '-----9', /* R(37106.) */  
5 ATTCOL_CUR PIC '-----9'; /* compute ATTCOL_CUR=0 */  
/* if R(37102.)>3 then ATTCOL_CUR=1 */
```

/* CREATION OF HIGHEST GRADE COMPLETED and ENROLLMENT STATUS AS OF 5-1-92 */

```

HGC_CUR=-4; ENROLL_CUR=-4;
do I=80 to LASTYR;
  if WEIGHT(I)=0 then do; HGCS(I)=-5; GRADE(I)=-5; end;
end;
if WEIGHT92 > 0 then do;
  if ASLI_CUR=0 & (HAVEDIP_CUR=-4 | HAVEDIP_CUR=0) then do; /* ATT NO */
    /* SEARCH */
    do I = LASTYR to 79 BY -1 WHILE(HGC_CUR = -4);
    if WEIGHT(I) > 0 then do; /* SEARCH INT YES */
      if INTMO(I)>=5 & (CURAT(I)=1 | (DLEMO(I)>=5 & DLEYR(I)=I)) &
        HGCS(I)>HGC(I) then HGC_CUR=HGCS(I); else HGC_CUR=HGC(I);
      end; /* SEARCH INT YES */
    end; /* SEARCH */
  end;
  if HGC_CUR>=12 then ENROLL_CUR=4;
  else if HGC_CUR>=0 then ENROLL_CUR=1;
end; /* ATT NO */
if ASLI_CUR=1 then do;
  if HGCS_CUR <=0 then do;
    if CURAT_CUR=1 & GRADE_CUR>=1 then do;
      HGC_CUR=GRADE_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;
    end;
    else if CURAT_CUR=0 & HGA_CUR>=1 then do;
      if DLEYR_CUR<LASTYR | DLEMO_CUR<5 then do;
        HGC_CUR=HGA_CUR;
        if HGC_CUR>=12 then ENROLL_CUR=4;
      end;
      else do; HGC_CUR=HGA_CUR-1;
        if HGC_CUR>=12 then ENROLL_CUR=4;
      end;
    end;
  end;
else do; HGC_CUR=HGA_CUR-1;
  if HGC_CUR>=12 then ENROLL_CUR=4;
end; /* ASLI = YES */
/* HGC_CUR STILL -4 */
/* CURAT and GRADE */
else ENROLL_CUR=2;
/* CURAT and GRADE */
/* NOTCURAT BUT GRADE */
/* DATES */
else ENROLL_CUR=1;
/* DATES */
else ENROLL_CUR=1;
/* DATES OK */
/* NOTCURAT BUT GRADE */
/* HGC_CUR STILL -4 */
/*ENROLL */
if HGCS_CUR=HGA_CUR | HGCS_CUR=GRADE_CUR then do;
  if (DLEYR_CUR=CURANTYR & DLEMO_CUR>=5) | (INTMO_CUR>=5 & CURAT_CUR=1) then do;
    HGC_CUR=HGCS_CUR-1;
    if HGC_CUR>=12 then ENROLL_CUR=3; else ENROLL_CUR=2;
  end;
  else HGC_CUR=HGCS_CUR; end;
if HGCS_CUR=HGA_CUR-1 | HGCS_CUR=GRADE_CUR-1 then HGC_CUR=HGCS_CUR;
else if (HGCS_CUR<HGA_CUR & HGA_CUR>0 & HGA_CUR<20) | (HGCS_CUR<GRADE_CUR &
  GRADE_CUR>0 & GRADE_CUR<20) then HGC_CUR=HGCS_CUR;
if ENROLL_CUR=-4 & CURAT_CUR=1 & HGC_CUR>=0 then do;
  if HGC_CUR>=12 then ENROLL_CUR=3; else ENROLL_CUR=2;
end;
if ENROLL_CUR=-4 & CURAT_CUR=0 & HGC_CUR>=0 then do;
  if HGC_CUR>=12 then ENROLL_CUR=4; else ENROLL_CUR=1;
end;
end; /* ASLI = YES */
if ((HGC_CUR=10 | HGC_CUR=11) & (HAVEDIP_CUR=1 | ATTCOL_CUR=1)) | (DIP_CUR=2 &
  HGC_CUR<12) | (HGC_CUR=-4 & (DIP_CUR=1 | DIP_CUR=3)) then do; HGC_CUR=12;
if ENROLL_CUR>0 then ENROLL_CUR=5-ENROLL_CUR;
else if CURAT_CUR=1 | (DLEYR_CUR=LASTYR & DLEMO_CUR>=5) then ENROLL_CUR=3;
else ENROLL_CUR=4;
end;
else if HGC_CUR<0 & DIP_CUR=-3 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>20 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;

```

Appendix 8: Highest Grade Completed & Enrollment Status Variable Creation

```
if HGC_CUR>HGC(LASTYR)+3 & HGC(LASTYR)>0 & DIP_CUR^=2 & DIP_CUR^=3 then do;
  HGC_CUR=-3; ENROLL_CUR=-3;
end;
if HGC_CUR>0 & HGC_CUR<HGC(LASTYR)-1 | (HGC(LASTYR)=12 & HGC_CUR=11) then do;
  HGC_CUR=-3; ENROLL_CUR=-3;
end;
end;                                         /*NEW */
TRU= 0;
do I=79 to LASTYR BY 1;
  if DIP(I)>1 then TRU =1;end;
  if TRU = 1 & HGC_CUR<12 then HGC_CUR=12;
  if TRU = 1 & (ENROLL_CUR=1 | ENROLL_CUR<0) then ENROLL_CUR=4;
  if HGC_CUR=-3 & ENROLL_CUR=-4 then ENROLL_CUR=-3;
else if HGC_CUR=-4 & ENROLL_CUR=-4 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;

/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected school */
/* variables. Values of -3 are recoded to valid values where auxilliary information indicates. However, there */
/* are a number of -3s computed for highest grade completed and enrollment status that remain. */
```

HGC_92=R(36569.) ENROLL_92=R(36570.)

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1993

```
/* The following code is applied to the data before the PLI program is executed. The purpose of the code is to      */
/* appropriately recode selected 1979 and 1980 variables.                                              */
/* if(R2283 eq 0 or R2282 eq 0) then do; DLEMO79=R2287; DLEYR79=R2288; end;                           */
/* else do; DLEMO79=R169; DLEYR79=R170; end;                                                 */
/* if(R4052 gt 0 and R2283 ne 0 and R2282 ne 0) then do; DLEMO80=R2287; DLEYR80=R2288; end;           */
/* if(R2275 eq 0 or R2276 eq 0) then GRADE79=R2277 ; else GRADE79=R2286;                         */
/* if(R2280 ge 0) then HGCS79=R2280; else compute HGCS79=R173; end if;                         */
/* if(R2276 eq 0 and R2286 gt 0) R156=1                                                       */
/*************************************************************
```

```
DCL 1 INREC79OLD,
 2 NORCID_OLD PIC '----9',
 2 PUBID_OLD PIC '----9',
 2 CURAT(79:LASTYR) PIC '----9', /* R( 156.) R( 2285.) R( 4168.) R( 6639.) R( 9053.) R(12052.)
                                  R(16045.) R(19050.) R(23059.) R(25084.) R(29075.) R(31096.) R(35096.) R(37096.) */
 2 DIP(79:LASTYR) PIC '----9', /* R( 183.) R( 2300.) R( 4182.) R( 6653.) R( 9067.) R(12066.)
                                 R(16059.) R(19061.) R(23070.) R(25095.) R(29086.) R(31107.) R(35107.) R(37107.) */
 2 DLEMO(79:LASTYR) PIC '----9', /* DLEMO79 DLEMO80 R( 4170.) R( 6641.) R( 9055.) R(12054.)
                                 R(16047.) R(19052.) R(23061.) R(25086.) R(29077.) R(31098.) R(35098.) R(37098.) */
 2 DLEYR(79:LASTYR) PIC '----9', /* DLEYR79 DLEYR80 R( 4171.) R( 6642.) R( 90546) R(12055.)
                                 R(16048.) R(19053.) R(23062.) R(25087.) R(29078.) R(31099.) R(35099.) R(37099.) */
 2 GRADE(79:LASTYR) PIC '----9', /* GRADE79 R( 2286.) R( 4169.) R( 6640.) R( 9054.) R(12053.)
                                 R(16046.) R(19051.) R(23060.) R(25085.) R(29076.) R(31097.) R(35097.) R(37097.) */
 2 HGA(79:LASTYR) PIC '----9', /* R( 172.) R( 2291.) R( 4173.) R( 6644.) R( 9058.) R(12057.)
                                 R(16050.) R(19055.) R(23064.) R(25089.) R(29080.) R(31101.) R(35101.) R(37101.) */
 2 HGC(79:LASTYR) PIC '----9', /* R( 2167.) R( 4064.) R( 6189.) R( 8982.) R(11450.) R(15202.)
                                 R(18909.) R(22580.) R(24454.) R(28711.) R(30748.) R(34015.) R(36569.) R(40074.) */
 2 HGCS(79:LASTYR) PIC '----9', /* HGCS79 R( 2292.) R( 4174.) R( 6645.) R( 9059.) R(12058.)
                                 R(16051.) R(19056.) R(23065.) R(25090.) R(29081.) R(31102.) R(35102.) R(37102.) */
 2 INTMO(79:LASTYR) PIC '----9', /* R( 1725.) R( 3292.) R( 5307.) R( 8099.) R(10457.) R(14275.)
                                 R(17946.) R(21562.) R(23657.) R(27425.) R(29861.) R(33025.) R(35734.) R(39176.) */
 2 WEIGHT(79:LASTYR) PIC '9999999'; /* R( 2161.) R( 4052.) R( 6146.) R( 8967.) R(11444.) R(15196.)
                                         R(18902.) R(22573.) R(24445.) R(28700.) R(30738.) R(34002.) R(36558.) R(40063.) */


```

```
DCL 1 INRECNEW,
 2 NORCID_CUR PIC '----9',
 2 INTOB_CUR PIC '9999999', /* DUMMY */
 2 SCHOOL_CUR,
   5 ASLI_CUR PIC '----9', /* R(41347.) */
   5 CURAT_CUR PIC '----9', /* R(41374.) */
   5 DIP_CUR PIC '----9', /* R(41385.) */
   5 DLEMO_CUR PIC '----9', /* R(41376.) */
   5 DLEYR_CUR PIC '----9', /* R(41376.01.) */
   5 GRA_CUR PIC '----9', /* compute GRA_CUR=-3 */ /* if R(41374.)=1 then GRA_CUR=R(41374.) */
   5 GRADE_CUR PIC '----9', /* else if R(41374.)=0 then GRA_CUR=R(41378.) */
   5 HGA_CUR PIC '----9', /* else if R(41347.)=0 then GRA_CUR=-4 */ /* R(41375.) */
   5 HGCS_CUR PIC '----9', /* R(41378.) */
   5 INTMO_CUR PIC '----9', /* R(41379.) */
   5 HAVEDIP_CUR PIC '----9', /* R(41002.) */
   5 ATTCOL_CUR PIC '----9'; /* R(41384.) */
   /* compute ATTCOL_CUR=0 */ /* if R(41379.)>3 then ATTCOL_CUR=1 */

```

/* CREATION OF HIGHEST GRADE COMPLETED and ENROLLMENT STATUS AS OF 5-1-93 */

```

HGC_CUR=-4; ENROLL_CUR=-4;
do I=80 to LASTYR;
  if WEIGHT(I)=0 then do; HGCS(I)=-5; GRADE(I)=-5; end;
end;
if ASLI_CUR=0 & (HAVEDIP_CUR=-4 | HAVEDIP_CUR=0) then do; /* ATT NO */
do I = LASTYR to 79 BY -1 WHILE(HGC_CUR = -4);           /* SEARCH */
  if WEIGHT(I) > 0 then do;                                /* SEA INT YES */
    if INTMO(I)>=5 & (CURAT(I)=1 | (DLEMO(I)>=5 & DLEYR(I)=I)) & HGCS(I)>HGC(I) then
      HGC_CUR=HGCS(I);
    else HGC_CUR=HGC(I);
  end;                                                 /* SEA INT YES */
end;                                                 /* SEARCH */
if HGC_CUR>=12 then ENROLL_CUR=4;
else if HGC_CUR>=0 then ENROLL_CUR=1;
end;                                         /* - ATT NO */
if ASLI_CUR=1 then do;
  if HGCS_CUR <=0 then do;                            /* ASLI = YES */
    if CURAT_CUR=1 & GRADE_CUR>=1 then do;          /* HGC_CUR STILL -4 */
      HGC_CUR=GRADE_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;               /* CURAT and GRADE */
    end;
  else if CURAT_CUR=0 & HGA_CUR>=1 then do;          /* NOTCURAT BUT GRADE */
    if DLEYR_CUR<LASTYR | DLEMO_CUR<5 then do;
      HGC_CUR=HGA_CUR;
      if HGC_CUR>=12 then ENROLL_CUR=4;               /* DATES */
    end;
  else do;
    HGC_CUR=HGA_CUR-1;
    if HGC_CUR>=12 then ENROLL_CUR=4;               /* DATES OK */
  end;
end;                                         /* NOTCURAT BUT GRADE */
else do;                                     /* HGC_CUR STILL -4 */
  if HGCS_CUR=HGA_CUR | HGCS_CUR=GRADE_CUR then do;
    if (DLEYR_CUR=CURANTYR & DLEMO_CUR>=5) | (INTMO_CUR>=5 & CURAT_CUR=1) then do;
      HGC_CUR=HGCS_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;               /* ENROLL */
    end;
  else HGC_CUR=HGCS_CUR;
end;
if HGCS_CUR=HGA_CUR-1 | HGCS_CUR=GRADE_CUR-1 then HGC_CUR=HGCS_CUR;
else if (HGCS_CUR<HGA_CUR & HGA_CUR>0 & HGA_CUR<20) | (HGCS_CUR<GRADE_CUR &
  GRADE_CUR>0 & GRADE_CUR<20) then HGC_CUR=HGCS_CUR;
if ENROLL_CUR=-4 & CURAT_CUR=1 & HGC_CUR>=0 then do;
  if HGC_CUR>=12 then ENROLL_CUR=3;               /* ENROLL */
end;
if ENROLL_CUR=-4 & CURAT_CUR=0 & HGC_CUR>=0 then do;
  if HGC_CUR>=12 then ENROLL_CUR=4;               /* ENROLL */
end;
end;                                         /* ASLI = YES */
if ((HGC_CUR=10 | HGC_CUR=11) & (HAVEDIP_CUR=1 | ATTCOL_CUR=1)) | (DIP_CUR=2 &
  HGC_CUR<12) | (HGC_CUR=-4 & (DIP_CUR=1 | DIP_CUR=3)) then do;
  HGC_CUR=12;
if ENROLL_CUR>0 then ENROLL_CUR=5-ENROLL_CUR;
else if CURAT_CUR=1 | (DLEYR_CUR=LASTYR & DLEMO_CUR>=5) then ENROLL_CUR=3;
else ENROLL_CUR=4;

```

```
end;
else if HGC_CUR<0 & DIP_CUR=-3 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>20 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>HGC(LASTYR)+3 & HGC(LASTYR)>0 & DIP_CUR^=2 & DIP_CUR^=3 then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;
if HGC_CUR>0 & HGC_CUR<HGC(LASTYR)-1 | (HGC(LASTYR)=12 & HGC_CUR=11) then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;
TRU= 0;
do I=79 to LASTYR BY 1;
    if DIP(I)>1 then TRU =1;end;
    if TRU = 1 & HGC_CUR<12 then HGC_CUR=12;
    if TRU = 1 & (ENROLL_CUR=1 | ENROLL_CUR<0) then ENROLL_CUR=4;
    if HGC_CUR=-3 & ENROLL_CUR=-4 then ENROLL_CUR=-3;
else if HGC_CUR=-4 & ENROLL_CUR=-4 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;

/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected school */
/* variables. Values of -3 are recoded to valid values where auxilliary information indicates. However , there */
/* are a number of -3s computed for highest grade completed and enrollment status that remain. */

/* HGC_93=R(44185.) */
/* ENROLL_93=R(44186.) */
```

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1994

```
/* The following code is applied to the data before the PLI program is executed. The purpose      */
/* of the code is to appropriately recode selected 1979 and 1980 variables.                      */
/* if (R2283 eq 0 or R2282 eq 0) then do; DLEMO79=R2287; DLEYR79=R2288;end;                  */
/* else do; DLEMO79=R169; DLEYR79=R170; end;                                              */
/* if (R4052 gt 0 and R2283 ne 0 and R2282 ne 0) then do; DLEMO80=R2287; DLEYR80=R2288; end; */
/* if (R2275 eq 0 or R2276 eq 0) then GRADE79=R2277 ; else GRADE79=R2286;                      */
/* if (R2280 ge 0) then HGCS79=R2280; else compute HGCS79=R173; end if;                         */
/* if (R2276 eq 0 and R2286 gt 0) R156=1                                                       */
/*********************************************************************
```

```
DCL 1 OUTREC,
 2 PUBID PIC '-----9',
 2 NORCID PIC '-----9',
 2 HGC_CUR PIC '-----9',
 2 ENROLL_CUR PIC '-----9';
 /* 2 CASEIDD CHAR(9), */

DCL 1 INREC79OLD,
 2 NORCID_OLD PIC '-----9',
 2 PUBID_OLD PIC '-----9',
 2 CURAT(79:LASTYR) PIC '-----9', /*R( 156.) R( 2285.) R( 4168.) R( 6639.) R( 9053.) R(12052.)
                                     R(16045.) R(19050.) R(23059.) R(25084.) R(29075.) R(31096.) R(35096.) R(37096.) R(41374.) */
 2 DIP(79:LASTYR) PIC '-----9', /*R( 183.) R( 2300.) R( 4182.) R( 6653.) R( 9067.) R(12066.)
                                     R(16059.) R(19061.) R(23070.) R(25095.) R(29086.) R(31107.) R(35107.) R(37107.) R(41384.)*/
 2 DLEMO(79:LASTYR) PIC '-----9', /*DLEMO79 DLEMO80 R( 4170.) R( 6641.) R( 9055.) R(12054.)
                                     R(16047.) R(19052.) R(23061.) R(25086.) R(29077.) R(31098.) R(35098.) R(37098.) R(41376.)*/
 2 DLEYR(79:LASTYR) PIC '-----9', /*DLEYR79 DLEYR80 R( 4171.) R( 6642.) R( 90546) R(12055.)
                                     R(16048.) R(19053.) R(23062.) R(25087.) R(29078.) R(31099.) R(35099.) R(37099.) R(41376.01)*/
 2 GRADE(79:LASTYR) PIC '-----9', /*GRADE79 R( 2286.) R( 4169.) R( 6640.) R( 9054.) R(12053.)
                                     R(16046.) R(19051.) R(23060.) R(25085.) R(29076.) R(31097.) R(35097.) R(37097.) R(41375.)*/
 2 HGA(79:LASTYR) PIC '-----9', /*R( 172.) R( 2291.) R( 4173.) R( 6644.) R( 9058.) R(12057.)
                                     R(16050.) R(19055.) R(23064.) R(25089.) R(29080.) R(31101.) R(35101.) R(37101.) R(41378.)*/
 2 HGC(79:LASTYR) PIC '-----9', /*R( 2167.) R( 4064.) R( 6189.) R( 8982.) R(11450.) R(15202.)
                                     R(18909.) R(22580.) R(24454.) R(28711.) R(30748.) R(34015.) R(36569.) R(40074.) R(44185.)*/
 2 HGCS(79:LASTYR) PIC '-----9', /*HGCS79 R( 2292.) R( 4174.) R( 6645.) R( 9059.) R(12058.)
                                     R(16051.) R(19056.) R(23065.) R(25090.) R(29081.) R(31102.) R(35102.) R(37102.) (R41379.)*/
 2 INTMO(79:LASTYR) PIC '-----9', /*R( 1725.) R( 3292.) R( 5307.) R( 8099.) R(10457.) R(14275.)
                                     R(17946.) R(21562.) R(23657.) R(27425.) R(29861.) R(33025.) R(35734.) R(39176.) R(41002.)*/
 2 WEIGHT(79:LASTYR) PIC '9999999'; /*R( 2161.) R( 4052.) R( 6146.) R( 8967.) R(11444.) R(15196.)
                                         R(18902.) R(22573.) R(24445.) R(28700.) R(30738.) R(34002.) R(36558.) R(40063.) R(44174.)*/
/* 2 CASEID CHAR(9), */
/* 2 RNICUR PIC '9999999'; - R(50805.) */

DCL 1 INRECNEW,
 2 NORCID_CUR PIC '-----9',
 2 INTOB_CUR PIC '9999999', /* DUMMY */
 2 SCHOOL_CUR,
 5 ASLI_CUR PIC '-----9', /* R(45233.) */
 5 CURAT_CUR PIC '-----9', /* R(45260.) */
 5 DIP_CUR PIC '-----9', /* R(45271.) */
 5 DLEMO_CUR PIC '-----9', /* R(45262.) */
 5 DLEYR_CUR PIC '-----9', /* R(45262.01) */
 5 GRA_CUR PIC '-----9', /* compute GRA_CUR=-3 */
 /* if R(45260.)=1 then GRA_CUR=R(45260.) */
 /* else if R(45260.)=0 then GRA_CUR=R(45264.) */
 /* else if R(45265.)=0 then GRA_CUR=-4 */
 5 GRADE_CUR PIC '-----9', /* R(45261.) */


```

```

5 HGA_CUR  PIC '-----9',      /* R(45264.) */
5 HGCS_CUR PIC '-----9',     /* R(45265.) */
5 INTMO_CUR PIC '-----9',    /* R(45002.) */
5 HAVEDIP_CUR PIC '-----9',   /* R(45270.) */
5 ATTCOL_CUR PIC '-----9';   /* compute ATTCOL_CUR=0          */
                             /* if R(45267.)>3 or R(45268.)>3 or      */
                             /* R(45278.)=1 then ATTCOL_CUR=1  */

/* CREATION OF HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF 5-1-94 */

HGC_CUR=-4; ENROLL_CUR=-4;
do I=80 to LASTYR;
  if WEIGHT(I)=0 then do; HGCS(I)=-5; GRADE(I)=-5; end;
end;
/*IN 94 OLD AND NEW FILES HAVE ONLY INTERVIEWS */
/* IF THERE IS A NEED TO UNCOMMENT LOOP BELOW */
/* IF RNICUR=-4 then do; - LOOP NEW */
if ASLI_CUR=0 & (HAVEDIP_CUR=-4 | HAVEDIP_CUR=0) then do; /* ATT NO */
do I = LASTYR to 79 BY -1 WHILE(HGC_CUR = -4);           /* SEARCH */
  if WEIGHT(I) > 0 then do;                                /* SEA INT YES */
    if INTMO(I)>=5 & (CURAT(I)=1 | (DLEMO(I)>=5 & DLEYR(I)=I)) &
       HGCS(I)>HGC(I) then HGC_CUR=HGCS(I);               else HGC_CUR=HGC(I);
    end;                                                 /* SEA INT YES */
  end;                                                 /* SEARCH */
  if HGC_CUR>=12 then ENROLL_CUR=4;
  else if HGC_CUR>=0 then ENROLL_CUR=1;
end; /* - ATT NO */
if ASLI_CUR=1 then do;                                     /*ASLI = YES */
  if HGCS_CUR <=0 then do;                               /* HGC_CUR STILL -4 */
    if CURAT_CUR=1 & GRADE_CUR>=1 then do;             /* CURAT and GRADE */
      HGC_CUR=GRADE_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;
    end;
    else if CURAT_CUR=0 & HGA_CUR>=1 then do;           /* NOTCURAT BUT GRADE */
      if DLEYR_CUR<LASTYR | DLEMO_CUR<5 then do;
        HGC_CUR=HGA_CUR;
        if HGC_CUR>=12 then ENROLL_CUR=4;
      end;
    else do;
      HGC_CUR=HGA_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=4;
    end;
  end;
else do;                                                 /* DATES */
  if HGCS_CUR=HGA_CUR | HGCS_CUR=GRADE_CUR then do;
    if (DLEYR_CUR=CURANTYR & DLEMO_CUR>=5) | (INTMO_CUR>=5 & CURAT_CUR=1) then
      do; HGC_CUR=HGCS_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;                   else ENROLL_CUR=2;
    end;
    else HGC_CUR=HGCS_CUR;
  end;
  if HGCS_CUR=HGA_CUR-1 | HGCS_CUR=GRADE_CUR-1 then HGC_CUR=HGCS_CUR;
  else if (HGCS_CUR<HGA_CUR & HGA_CUR>0 & HGA_CUR<20) | (HGCS_CUR<GRADE_CUR &
    GRADE_CUR>0 & GRADE_CUR<20) then HGC_CUR=HGCS_CUR;
  if ENROLL_CUR=-4 & CURAT_CUR=1 & HGC_CUR>=0 then do;
    if HGC_CUR>=12 then ENROLL_CUR=3;                   else ENROLL_CUR=2;
  end;
end;

```

```
end;
if ENROLL_CUR=-4 & CURAT_CUR=0 & HGC_CUR>=0 then do;
    if HGC_CUR>=12 then ENROLL_CUR=4; else ENROLL_CUR=1;
end;
end;
end; /* ASLI = YES */
if ((HGC_CUR=10 | HGC_CUR=11) & (HAVEDIP_CUR=1 | ATTCOL_CUR=1)) |(DIP_CUR=2 &
    HGC_CUR<12) | (HGC_CUR=-4 & (DIP_CUR=1 | DIP_CUR=3)) then do;
    HGC_CUR=12;
if ENROLL_CUR>0 then ENROLL_CUR=5-ENROLL_CUR;
else if CURAT_CUR=1 | (DLEYR_CUR=LASTYR & DLEMO_CUR>=5) then ENROLL_CUR=3;
else ENROLL_CUR=4;
end;
else if HGC_CUR<0 & DIP_CUR=-3 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>20 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>HGC(LASTYR)+3 & HGC(LASTYR)>0 & DIP_CUR^=2 & DIP_CUR^=3 then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;
if HGC_CUR>0 & HGC_CUR<HGC(LASTYR)-1 | (HGC(LASTYR)=12 & HGC_CUR=11) then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;
/* end; LOOP NEW IN 94 OLD AND NEW FILES HAVE ONLY INTERVIEWS */
/* IF THERE IS A NEED TO UNCOMMENT THIS END */
TRU= 0;
do I=79 to LASTYR BY 1;
if DIP(I)>1 then TRU =1;end;
if TRU = 1 & HGC_CUR<12 then HGC_CUR=12;
if TRU = 1 & (ENROLL_CUR=1 | ENROLL_CUR<0) then ENROLL_CUR=4;
if HGC_CUR=-3 & ENROLL_CUR=-4 then ENROLL_CUR=-3;
else if HGC_CUR=-4 & ENROLL_CUR=-4 then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;

/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected */
/* school variables. Values of -3 are recoded to valid values where auxilliary information indicates. However, */
/* there are a number of -3s computed for highest grade completed and enrollment status that remain. */

/* HGC_94=R(50815.) */ /* ENROLL_94=R(50816.) */
```

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1996

```

HGC_CUR=-4;
ENROLL_CUR=-4;
do I=2 to LASTRND;
  if WEIGHT(I)=0 then do; HGCS(I)=-5; GRADE(I)=-5; end;
end;
/*IN 96 OLD AND NEW FILES HAVE ONLY INTERVIEWS */
/* IF THERE IS A NEED TO UNCOMMENT LOOP BELOW */
/* if RNICUR=-4 then do;      - LOOP NEW */
if ASLI_CUR=0 & (HAVEDIP_CUR=-4 | HAVEDIP_CUR=0) then do; /* ATT NO */
do I = LASTRND to 1 BY -1 WHILE(HGC_CUR = -4);           /* SEARCH */
  if WEIGHT(I) > 0 then do;                                /* SEA INT YES */
    if INTMO(I)>=5 & (CURAT(I)=1 | (DLEMO(I)>=5 & DLEYR(I)=INTYR(I))) &
       HGCS(I)>HGC(I) then HGC_CUR=HGCS(I);               else HGC_CUR=HGC(I);
    end;                                                 /* SEA INT YES */
  end;                                                 /* SEARCH */
  if HGC_CUR>=12 then ENROLL_CUR=4; else if HGC_CUR>=0 then ENROLL_CUR=1;
end;                                                 /* - ATT NO */
if ASLI_CUR=1 then do;                                /* ASLI = YES */
  if HGCS_CUR <=0 then do;
    if CURAT_CUR=1 & GRADE_CUR>=1 then do;
      HGC_CUR=GRADE_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;
    end;
    else if CURAT_CUR=0 & HGA_CUR>=1 then do;
      if DLEYR_CUR<LASTYR | DLEMO_CUR<5 then do;
        HGC_CUR=HGA_CUR;
        if HGC_CUR>=12 then ENROLL_CUR=4;
      end;
    else do;
      HGC_CUR=HGA_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=4;
    end;
  end;
else do;
  if HGCS_CUR=HGA_CUR | HGCS_CUR=GRADE_CUR then do;
    if (DLEYR_CUR=CURANTYR & DLEMO_CUR>=5) | (INTMO_CUR>=5 & CURAT_CUR=1) then
      do; HGC_CUR=HGCS_CUR-1;
      if HGC_CUR>=12 then ENROLL_CUR=3;                   else ENROLL_CUR=2;
    end;
    else HGC_CUR=HGCS_CUR;
  end;
  if HGCS_CUR=HGA_CUR-1 | HGCS_CUR=GRADE_CUR-1 then HGC_CUR=HGCS_CUR;
  else if (HGCS_CUR<HGA_CUR & HGA_CUR>0 & HGA_CUR<20) | (HGCS_CUR<GRADE_CUR &
    GRADE_CUR>0 & GRADE_CUR<20) then HGC_CUR=HGCS_CUR;
  if ENROLL_CUR=-4 & CURAT_CUR=1 & HGC_CUR>=0 then do;
    if HGC_CUR>=12 then ENROLL_CUR=3;                   else ENROLL_CUR=2;
  end;
  if ENROLL_CUR=-4 & CURAT_CUR=0 & HGC_CUR>=0 then do;
    if HGC_CUR>=12 then ENROLL_CUR=4;                   else ENROLL_CUR=1;
  end;
end;                                                 /* ASLI = YES */
if ((HGC_CUR=10 | HGC_CUR=11) & (HAVEDIP_CUR=1 | ATTCOL_CUR=1)) | (DIP_CUR=2 &
  HGC_CUR<12) | (HGC_CUR=-4 & (DIP_CUR=1 | DIP_CUR=3)) then do;
  HGC_CUR=12;

```

Appendix 8: Highest Grade Completed & Enrollment Status Variable Creation

```
if ENROLL_CUR>0 then ENROLL_CUR=5-ENROLL_CUR;
else if CURAT_CUR=1 | (DLEYR_CUR=LASTYR & DLEMO_CUR>=5) then ENROLL_CUR=3;
else ENROLL_CUR=4;
end;
else if HGC_CUR<0 & DIP_CUR=-3 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>20 then do; HGC_CUR=-3; ENROLL_CUR=-3; end;
if HGC_CUR>HGC(LASTRND)+3 & HGC(LASTRND)>0 & DIP_CUR^=2 & DIP_CUR^=3 then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;
if HGC_CUR>0 & HGC_CUR<HGC(LASTRND)-1 | (HGC(LASTRND)=12 & HGC_CUR=11) then do;
    HGC_CUR=-3; ENROLL_CUR=-3;
end;
/* end; LOOP NEW IN 96 OLD AND NEW FILES HAVE ONLY INTERVIEWS */
/*      IF THERE IS A NEED TO UNCOMMENT THIS END      */
TRU= 0;
do I=1 to LASTRND BY 1;
if DIP(I)>1 then TRU =1;end;
if TRU = 1 & HGC_CUR<12 then HGC_CUR=12;
if TRU = 1 & (ENROLL_CUR=1 | ENROLL_CUR<0) then ENROLL_CUR=4;
if HGC_CUR=-3 & ENROLL_CUR=-4 then ENROLL_CUR=-3;
else if HGC_CUR=-4 & ENROLL_CUR=-4 then do;
    HGC_CUR=-3;
    ENROLL_CUR=-3;
end;

/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected */
/* school variables. Values of -3 are recoded to valid values where auxilliary information indicates.*/
/* However, there are a number of -3s computed for highest grade completed and enrollment status that */
/* remain.          */
/* HGC_96=R(51668.)  */ /* ENROLL_96=R(51669.) */
```

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 1998

NOTE: The PL/I code used to create Highest Grade Completed and Enrollment Status variables through the 1996 release was converted to SPSS code for 1998 and subsequent releases.

```

/* The following code is applied to the data before the PLI program is executed. The purpose of      */
/* the code is to appropriately recode selected 1979 and 1980 variables.                         */
/* if (R2283 eq 0 or R2282 eq 0) then do dlemo79=R2287 dleyr79=R2288 end                      */
/* else do dlemo79=R169 dleyr79=R170 end                                                       */
/* if (R4052 gt 0 and R2283 ne 0 and R2282 ne 0) then do dlemo80=R2287 dleyr80=R2288 end      */
/* if (R2275 eq 0 or R2276 eq 0) then GRADE79=R2277 else GRADE79=R2286                         */
/* if (R2280 ge 0) then hgcs79=R2280 else compute hgcs79=R173                                */
/* ENDIF                                            */
/* if (R2276 eq 0 and R2286 gt 0) R156=1          */
/************************************************************* */

/* CREATE HIGHEST GRADE COMPLETED and ENROLLMENT STATUS AS OF 5-1-1998 */

compute HGC_CUR=-4
compute ENR_CUR=-4

do repeat HGCS=HGCS2 to HGCS17
    /GRADE=GRADE2 to GRADE17
    /WEIGHT=WEIGHT2 to WEIGHT17
    . do if (WEIGHT eq 0)
    . compute HGCS=-5
    . compute GRADE=-5
    . end if
end repeat PRINT

do if (NORCID eq 146944)
compute HGCS17=-5
compute GRADE17=-5
end if

compute WGT1B=WEIGHT17
compute WGT4B=WEIGHT14
compute WGT7B=WEIGHT11
compute WGT10B=WEIGHT8
compute WGT13B=WEIGHT5
compute WGT16B=WEIGHT2
compute WGT2B=WEIGHT16
compute WGT5B=WEIGHT13
compute WGT8B=WEIGHT10
compute WGT11B=WEIGHT7
compute WGT14B=WEIGHT4
compute WGT17B=WEIGHT1
compute WGT3B=WEIGHT15
compute WGT6B=WEIGHT12
compute WGT9B=WEIGHT9
compute WGT12B=WEIGHT6
compute WGT15B=WEIGHT3

compute INTMO1B=INTMO17
compute INTMO4B=INTMO14
compute INTMO7B=INTMO11
compute INTMO10B=INTMO8
compute INTMO13B=INTMO5
compute INTMO16B=INTMO2
compute INTMO2B=INTMO16
compute INTMO5B=INTMO13
compute INTMO8B=INTMO10
compute INTMO11B=INTMO7
compute INTMO14B=INTMO4
compute INTMO17B=INTMO1
compute INTMO3B=INTMO15
compute INTMO6B=INTMO12
compute INTMO9B=INTMO9
compute INTMO12B=INTMO6
compute INTMO15B=INTMO3

compute INTYR1B=INTYR17
compute INTYR4B=INTYR14
compute INTYR7B=INTYR11
compute INTYR10B=INTYR8
compute INTYR13B=INTYR5
compute INTYR16B=INTYR2
compute INTYR2B=INTYR16
compute INTYR5B=INTYR13
compute INTYR8B=INTYR10
compute INTYR11B=INTYR7
compute INTYR14B=INTYR4
compute INTYR17B=INTYR1
compute INTYR3B=INTYR15
compute INTYR6B=INTYR12
compute INTYR9B=INTYR9
compute INTYR12B=INTYR6
compute INTYR15B=INTYR3

compute CURAT1B=CURAT17
compute CURAT4B=CURAT14
compute CURAT2B=CURAT16
compute CURAT5B=CURAT13
compute CURAT3B=CURAT15
compute CURAT6B=CURAT12

```

Appendix 8: Highest Grade Completed & Enrollment Status Variable Creation

```

compute CURAT7B=CURAT11          compute CURAT8B=CURAT10          compute CURAT9B=CURAT9
compute CURAT10B=CURAT8          compute CURAT11B=CURAT7          compute CURAT12B=CURAT6
compute CURAT13B=CURAT5          compute CURAT14B=CURAT4          compute CURAT15B=CURAT3
compute CURAT16B=CURAT2          compute CURAT17B=CURAT1

compute DLEMO1B=DLEMO17          compute DLEMO2B=DLEMO16          compute DLEMO3B=DLEMO15
compute DLEMO4B=DLEMO14          compute DLEMO5B=DLEMO13          compute DLEMO6B=DLEMO12
compute DLEMO7B=DLEMO11          compute DLEMO8B=DLEMO10          compute DLEMO9B=DLEMO9
compute DLEMO10B=DLEMO8          compute DLEMO11B=DLEMO7          compute DLEMO12B=DLEMO6
compute DLEMO13B=DLEMO5          compute DLEMO14B=DLEMO4          compute DLEMO15B=DLEMO3
compute DLEMO16B=DLEMO2          compute DLEMO17B=DLEMO1

compute DLEYR1B=DLEYR17          compute DLEYR2B=DLEYR16          compute DLEYR3B=DLEYR15
compute DLEYR4B=DLEYR14          compute DLEYR5B=DLEYR13          compute DLEYR6B=DLEYR12
compute DLEYR7B=DLEYR11          compute DLEYR8B=DLEYR10          compute DLEYR9B=DLEYR9
compute DLEYR10B=DLEYR8          compute DLEYR11B=DLEYR7          compute DLEYR12B=DLEYR6
compute DLEYR13B=DLEYR5          compute DLEYR14B=DLEYR4          compute DLEYR15B=DLEYR3
compute DLEYR16B=DLEYR2          compute DLEYR17B=DLEYR1

compute HGCS1B=HGCS17          compute HGCS2B=HGCS16          compute HGCS3B=HGCS15
compute HGCS4B=HGCS14          compute HGCS5B=HGCS13          compute HGCS6B=HGCS12
compute HGCS7B=HGCS11          compute HGCS8B=HGCS10          compute HGCS9B=HGCS9
compute HGCS10B=HGCS8          compute HGCS11B=HGCS7          compute HGCS12B=HGCS6
compute HGCS13B=HGCS5          compute HGCS14B=HGCS4          compute HGCS15B=HGCS3
compute HGCS16B=HGCS2          compute HGCS17B=HGCS1

compute HGC1B=HGC17          compute HGC2B=HGC16          compute HGC3B=HGC15
compute HGC4B=HGC14          compute HGC5B=HGC13          compute HGC6B=HGC12
compute HGC7B=HGC11          compute HGC8B=HGC10          compute HGC9B=HGC9
compute HGC10B=HGC8          compute HGC11B=HGC7          compute HGC12B=HGC6
compute HGC13B=HGC5          compute HGC14B=HGC4          compute HGC15B=HGC3
compute HGC16B=HGC2          compute HGC17B=HGC1

do repeat WEIGHT=WGT1B to WGT17B /* ASLI eq NO */
    /INTMO=INTMO1B to INTMO17B
    /CURAT=CURAT1B to CURAT17B
    /DLEMO=DLEMO1B to DLEMO17B
    /DLEYR=DLEYR1B to DLEYR17B
    /INTYR=INTYR1B to INTYR17B
    /HGCS=HGCS1B to HGCS17B
    /HGC=HGC1B to HGC17B
    . LOOP if (ASLI_CUR eq 0 and (HVDIP_C eq -4 or HVDIP_C eq 0) and FLAG98 eq 1)
        . do if (HGC_CUR eq -4 and WEIGHT gt 0 and INTMO ge 5 and (CURAT eq 1 or (DLEMO ge 5 and DLEYR
            eq INTYR)) and HGCS gt HGC)           . compute HGC_CUR=HGCS
        . else if (HGC_CUR eq -4 and WEIGHT gt 0)      . compute HGC_CUR=HGC
        . end if
    . end LOOP if (WEIGHT gt 0 and HGC_CUR ne -4 and FLAG98 eq 0)
end repeat PRINT
do if (HGC_CUR ge 12) compute ENR_CUR=4
else if (HGC_CUR ge 0) compute ENR_CUR=1
end if

do if (ASLI_CUR eq 1 and FLAG98 eq 1 and HGCS_CUR le 0)          /*ASLI eq YES */
    . do if (CURAT_C eq 1 and GRADE_C ge 1)          /*HGC_CUR STILL -4 */
    . compute HGC_CUR=(GRADE_C - 1)                  /*CURAT and GRADE */
    . do if (HGC_CUR ge 12) . compute ENR_CUR=3

```

```

.     else          .     compute ENR_CUR=2
.     end if
.   else if (CURAT_C eq 0 and HGA_CUR ge 1)           /*NOT CURAT BUT GRADE */
.     do if (DLEYR_C lt LASTYR or DLEMO_C lt 5)        /* DATES */
.       compute HGC_CUR=HGA_CUR
.       do if (HGC_CUR ge 12)  .     compute ENR_CUR=4
.       else          .     compute ENR_CUR=1
.       end if
.     end if
.   else
.     compute HGC_CUR=(HGA_CUR - 1)
.     do if (HGC_CUR ge 12)  .     compute ENR_CUR=4
.     else          .     compute ENR_CUR=1
.   end if
.   end if /* DATES */
. else
.   compute HGC_CUR=(HGA_CUR - 1)
.   do if (HGC_CUR ge 12)  .     compute ENR_CUR=4
.   else          .     compute ENR_CUR=1
.   end if /* DATES OK */
. end if /* NOTCURAT BUT GRADE */ /* HGC_CUR STILL -4 */
else if (ASLI_CUR eq 1 and FLAG98 eq 1 and (HGCS_CUR eq HGA_CUR or HGCS_CUR eq GRADE_C)) /* ENROLL */
.   do if ((DLEYR_C eq CURANTYR and DLEMO_C ge 5) or (INTMO_C ge 5 and CURAT_C eq 1))
.     compute HGC_CUR=(HGCS_CUR - 1)
.     do if (HGC_CUR ge 12)  .     compute ENR_CUR=3
.     else          .     compute ENR_CUR=2
.   end if
. else
.   compute HGC_CUR=HGCS_CUR
.   do if (HGC_CUR ge 12)  .     compute ENR_CUR=4
.   else          .     compute ENR_CUR=1
.   end if
. end if
else if (ASLI_CUR eq 1 and FLAG98 eq 1 and (HGCS_CUR eq (HGA_CUR-1) or HGCS_CUR eq (GRADE_C-1)))
compute HGC_CUR=HGCS_CUR
. do if (ENR_CUR eq -4 and CURAT_C eq 1 and HGC_CUR ge 0)
. do if (HGC_CUR ge 12)  .     compute ENR_CUR=3
. else          .     compute ENR_CUR=2
. end if
. else if (ENR_CUR eq -4 and CURAT_C eq 0 and HGC_CUR ge 0)
. do if (HGC_CUR ge 12)  .     compute ENR_CUR=4
. else          .     compute ENR_CUR=1
. end if
. end if
else if (ASLI_CUR eq 1 and FLAG98 eq 1 and ((HGCS_CUR lt HGA_CUR and HGA_CUR gt 0 and
HGA_CUR lt 20) or (HGCS_CUR lt GRADE_C and GRADE_C gt 0 and GRADE_C lt 20)))
compute HGC_CUR=HGCS_CUR
. do if (ENR_CUR eq -4 and CURAT_C eq 1 and HGC_CUR ge 0)
. do if (HGC_CUR ge 12)  .     compute ENR_CUR=3
. else          .     compute ENR_CUR=2
. end if
. else if (ENR_CUR eq -4 and CURAT_C eq 0 and HGC_CUR ge 0)
. do if (HGC_CUR ge 12)  .     compute ENR_CUR=4
. else          .     compute ENR_CUR=1
. end if
. end if
end if /* ASLI eq YES */

```

FREQUENCIES VARIABLES=HGC_CUR ENR_CUR /FORMAT=ONEPAGE

Appendix 8: Highest Grade Completed & Enrollment Status Variable Creation

```
do if (((HGC_CUR eq 10 or HGC_CUR eq 11) and (HVDIP_C eq 1 or ATTCOL_C eq 1)) or (DIP_CUR eq 2  
and HGC_CUR lt 12) or (HGC_CUR eq -4 and (DIP_CUR eq 1 or DIP_CUR eq 3)))  
compute HGC_CUR=12  
. do if (ENR_CUR gt 0) . compute ENR_CUR=(5-ENR_CUR)  
. else if (CURAT_C eq 1 or (DLEYR_C eq LASTYR and DLEMO_C ge 5)). compute ENR_CUR=3  
. else . compute ENR_CUR=4  
. end if  
else if (HGC_CUR lt 0 and DIP_CUR eq -3)  
compute HGC_CUR=-3  
compute ENR_CUR=-3  
end if  
  
do if (HGC_CUR gt 20)  
compute HGC_CUR=-3  
compute ENR_CUR=-3  
end if  
  
do if (HGC_CUR gt (HGC_DLI + 3) and HGC_DLI gt 0 and DIP_CUR ne 2 and DIP_CUR ne 3)  
compute HGC_CUR=-3  
compute ENR_CUR=-3  
end if  
  
do if ((HGC_CUR gt 0 and HGC_CUR lt (HGC_DLI - 1)) or (HGC_DLI eq 12 and HGC_CUR eq 11))  
compute HGC_CUR=-3  
compute ENR_CUR=-3  
end if  
  
/* END LOOP NEW IN 96 OLD AND NEW FILES HAVE ONLY INTERVIEWS */  
/* IF THERE IS A NEED TO UNCOMMENT THIS END */  
  
compute TRU=0  
  
do repeat DIP=DIP1 to DIP17  
if (DIP gt 1) TRU=1  
if (TRU eq 1 and HGC_CUR lt 12) HGC_CUR=12  
if (TRU eq 1 and (ENR_CUR eq 1 or ENR_CUR lt 0)) ENR_CUR=4  
if (HGC_CUR eq -3 and ENR_CUR eq -4) ENR_CUR=-3  
end repeat  
  
do if (FLAG98 eq 0)  
compute HGC_CUR=-5  
compute ENR_CUR=-5  
end if  
  
do if (HGC_CUR eq -4 and FLAG98 eq 1)  
compute HGC_CUR=-3  
end if  
  
do if (ENR_CUR eq -4 and FLAG98 eq 1)  
compute ENR_CUR=-3  
end if  
  
/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected school  
variables. Values of -3 are recoded to valid values where auxilliary information indicates. However, there  
are a number of -3s computed for highest grade completed and enrollment status that remain. */
```

HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF MAY 1, 2000

NOTE: The PL/I code used to create Highest Grade Completed and Enrollment Status variables through the 1996 release was converted to SPSS code for 1998 and subsequent releases.

```

/* The following code is applied to the data before the PLI program is executed. The purpose of      */
/* the code is to appropriately recode selected 1979 and 1980 variables.                         */
/* if (R2283 eq 0 or R2282 eq 0) then do dlemo79=R2287 dleyr79=R2288 end                      */
/* else do dlemo79=R169 dleyr79=R170 end                                                       */
/* if (R4052 gt 0 and R2283 ne 0 and R2282 ne 0) then do dlemo80=R2287 dleyr80=R2288 end      */
/* if (R2275 eq 0 or R2276 eq 0) then GRADE79=R2277 else GRADE79=R2286                         */
/* if (R2280 ge 0) then hgcs79=R2280 else compute hgcs79=R173                                */
/* ENDIF                                            */
/* if (R2276 eq 0 and R2286 gt 0) R156=1          */
/************************************************************* */

/* CREATE HIGHEST GRADE COMPLETED AND ENROLLMENT STATUS AS OF 5-1-2000 */

compute hgc_cur=-4           compute enr_cur=-4

do repeat hgcs=hgcs2 to hgcs18      /grade=grade2 to grade18      /weight=weight2 to weight18
. do if (weight eq 0)   . compute hgcs=-5   . compute grade=-5   . end if
end repeat print

compute wgt1b=weight18          compute wgt2b=weight17          compute wgt3b=weight16
compute wgt4b=weight15          compute wgt5b=weight14          compute wgt6b=weight13
compute wgt7b=weight12          compute wgt8b=weight11          compute wgt9b=weight10
compute wgt10b=weight9          compute wgt11b=weight8          compute wgt12b=weight7
compute wgt13b=weight6          compute wgt14b=weight5          compute wgt15b=weight4
compute wgt16b=weight3          compute wgt17b=weight2          compute wgt18b=weight1

compute intmo1b=intmo18         compute intmo2b=intmo17         compute intmo3b=intmo16
compute intmo4b=intmo15         compute intmo5b=intmo14         compute intmo6b=intmo13
compute intmo7b=intmo12         compute intmo8b=intmo11         compute intmo9b=intmo10
compute intmo10b=intmo9          compute intmo11b=intmo8         compute intmo12b=intmo7
compute intmo13b=intmo6          compute intmo14b=intmo5         compute intmo15b=intmo4
compute intmo16b=intmo3          compute intmo17b=intmo2         compute intmo18b=intmo1

compute intyr1b=intyr18          compute intyr2b=intyr17         compute intyr3b=intyr16
compute intyr4b=intyr15          compute intyr5b=intyr14         compute intyr6b=intyr13
compute intyr7b=intyr12          compute intyr8b=intyr11         compute intyr9b=intyr10
compute intyr10b=intyr9          compute intyr11b=intyr8         compute intyr12b=intyr7
compute intyr13b=intyr6          compute intyr14b=intyr5         compute intyr15b=intyr4
compute intyr16b=intyr3          compute intyr17b=intyr2         compute intyr18b=intyr1

compute curat1b=curat18          compute curat2b=curat17         compute curat3b=curat16
compute curat4b=curat15          compute curat5b=curat14         compute curat6b=curat13
compute curat7b=curat12          compute curat8b=curat11         compute curat9b=curat10
compute curat10b=curat9          compute curat11b=curat8         compute curat12b=curat7
compute curat13b=curat6          compute curat14b=curat5         compute curat15b=curat4
compute curat16b=curat3          compute curat17b=curat2         compute curat18b=curat1

compute dlemo1b=dlemo18          compute dlemo2b=dlemo17         compute dlemo3b=dlemo16
compute dlemo4b=dlemo15          compute dlemo5b=dlemo14         compute dlemo6b=dlemo13
compute dlemo7b=dlemo12          compute dlemo8b=dlemo11         compute dlemo9b=dlemo10
compute dlemo10b=dlemo9          compute dlemo11b=dlemo8         compute dlemo12b=dlemo7
compute dlemo13b=dlemo6          compute dlemo14b=dlemo5         compute dlemo15b=dlemo4
compute dlemo16b=dlemo3          compute dlemo17b=dlemo2         compute dlemo18b=dlemo1

```

```

compute dleyr1b=dleyr18          compute dleyr2b=dleyr17          compute dleyr3b=dleyr16
compute dleyr4b=dleyr15          compute dleyr5b=dleyr14          compute dleyr6b=dleyr13
compute dleyr7b=dleyr12          compute dleyr8b=dleyr11          compute dleyr9b=dleyr10
compute dleyr10b=dleyr9          compute dleyr11b=dleyr8          compute dleyr12b=dleyr7
compute dleyr13b=dleyr6          compute dleyr14b=dleyr5          compute dleyr15b=dleyr4
compute dleyr16b=dleyr3          compute dleyr17b=dleyr2          compute dleyr18b=dleyr1

compute hgcs1b=hgcs18          compute hgcs2b=hgcs17          compute hgcs3b=hgcs16
compute hgcs4b=hgcs15          compute hgcs5b=hgcs14          compute hgcs6b=hgcs13
compute hgcs7b=hgcs12          compute hgcs8b=hgcs11          compute hgcs9b=hgcs10
compute hgcs10b=hgcs9          compute hgcs11b=hgcs8          compute hgcs12b=hgcs7
compute hgcs13b=hgcs6          compute hgcs14b=hgcs5          compute hgcs15b=hgcs4
compute hgcs16b=hgcs3          compute hgcs17b=hgcs2          compute hgcs18b=hgcs1

compute hgc1b=hgc18          compute hgc2b=hgc17          compute hgc3b=hgc16
compute hgc4b=hgc15          compute hgc5b=hgc14          compute hgc6b=hgc13
compute hgc7b=hgc12          compute hgc8b=hgc11          compute hgc9b=hgc10
compute hgc10b=hgc9          compute hgc11b=hgc8          compute hgc12b=hgc7
compute hgc13b=hgc6          compute hgc14b=hgc5          compute hgc15b=hgc4
compute hgc16b=hgc3          compute hgc17b=hgc2          compute hgc18b=hgc1

do repeat weight=wgt1b to wgt18b
    /intmo=intmo1b to intmo18b
    /dlemo=dlemo1b to dlemo18b
    /intyr=intyr1b to intyr18b
    /hgc=hgc1b to hgc18b
    /* asli eq no */
    /curat=curat1b to curat18b
    /dleyr=dleyr1b to dleyr18b
    /hgcs=hgcs1b to hgcs18b
    /* asli eq yes */ /* hgc_cur still -4 */
    /* curat and grade */
    /* dates */

.loop if (asli_cur eq 0 and (hvdpip_c eq -4 or hvdpip_c eq 0) and flag00 eq 1)
. do if (hgc_cur eq -4 and weight gt 0 and intmo ge 5 and (curat eq 1 or (dlemo ge 5 and dleyr eq intyr)) and hgcs
      gt hgc)           . compute hgc_cur=hgcs
. else if (hgc_cur eq -4 and weight gt 0)   . compute hgc_cur=hgc
. end if
.end loop if (weight gt 0 and hgc_cur ne -4 and flag00 eq 0)
end repeat print

do if (hgc_cur ge 12)          compute enr_cur=4
else if (hgc_cur ge 0)         compute enr_cur=1
                                end if
                                /* dates ok */

do if (asli_cur eq 1 and flag00 eq 1 and hgcs_cur le 0)          /* asli eq yes */ /* hgc_cur still -4 */
. do if (curat_c eq 1 and grade_c ge 1)                          /* curat and grade */
. compute hgc_cur=(grade_c - 1)
. do if (hgc_cur ge 12)           . compute enr_cur=3
. else                         . compute enr_cur=2
. else if (curat_c eq 0 and hga_cur ge 1)                      . end if
. do if (dleyr_c lt lastyr or dlemo_c lt 5)                   /* not curat but grade */
. do if (hgc_cur ge 12)           . compute hgc_cur=hga_cur /* dates */
. else                         . compute enr_cur=4
. else                         . compute enr_cur=1
. end if
. else                           . compute hgc_cur=(hga_cur - 1)
. do if (hgc_cur ge 12)           . compute enr_cur=4
. else                           . compute enr_cur=1
. end if
. end if
/* notcurat but grade */ /* hgc_cur still -4 */
else if (asli_cur eq 1 and flag00 eq 1 and (hgcs_cur eq hga_cur or hgcs_cur eq grade_c)) /* enroll */
. do if ((dleyr_c eq curantyr and dlemo_c ge 5) or (intmo_c ge 5 and curat_c eq 1))
. compute hgc_cur=(hgcs_cur - 1)
. do if (hgc_cur ge 12)           . compute enr_cur=3

```

```

. else compute enr_cur=2 . end if
. else
. compute hgc_cur=hgcs_cur
. do if (hgc_cur ge 12) . compute enr_cur=4
. else . compute enr_cur=1 . end if
. end if
else if (asli_cur eq 1 and flag00 eq 1 and (hgcs_cur eq (hga_cur-1) or hgcs_cur eq (grade_c-1)))
    compute hgc_cur=hgcs_cur
do if (enr_cur eq -4 and curat_c eq 1 and hgc_cur ge 0)
. do if (hgc_cur ge 12) . compute enr_cur=3
. else . compute enr_cur=2 . end if
. else if (enr_cur eq -4 and curat_c eq 0 and hgc_cur ge 0)
. do if (hgc_cur ge 12) . compute enr_cur=4
. else . compute enr_cur=1 . end if
. end if
else if (asli_cur eq 1 and flag00 eq 1 and ((hgcs_cur lt hga_cur and hga_cur gt 0 and hga_cur lt 20) or (hgcs_cur lt
    grade_c and grade_c gt 0 and grade_c lt 20))) compute hgc_cur=hgcs_cur
do if (enr_cur eq -4 and curat_c eq 1 and hgc_cur ge 0)
. do if (hgc_cur ge 12) . compute enr_cur=3
. else . compute enr_cur=2 . end if
. else if (enr_cur eq -4 and curat_c eq 0 and hgc_cur ge 0)
. do if (hgc_cur ge 12) . compute enr_cur=4
. else . compute enr_cur=1 . end if
. end if
end if /* asli eq yes */

do if (((hgc_cur eq 10 or hgc_cur eq 11) and (hvdpip_c eq 1 or attcol_c eq 1)) or (dip_cur eq 2 and hgc_cur lt 12) or
    (hgc_cur eq -4 and (dip_cur eq 1 or dip_cur eq 3))) compute hgc_cur=12
do if (enr_cur gt 0) . compute enr_cur=(5-enr_cur)
else if (curat_c eq 1 or (dleyr_c eq lastyr and dlemo_c ge 5)) . compute enr_cur=3
else . compute enr_cur=4 . end if
else if (hgc_cur lt 0 and dip_cur eq -3) compute hgc_cur=-3
compute enr_cur=-3 end if
do if (hgc_cur gt 20) compute hgc_cur=-3
compute enr_cur=-3 end if
do if (hgc_cur gt (hgc_dli + 3) and hgc_dli gt 0 and dip_cur ne 2 and dip_cur ne 3)
    compute hgc_cur=-3
    compute enr_cur=-3 end if
do if ((hgc_cur gt 0 and hgc_cur lt (hgc_dli - 1)) or (hgc_dli eq 12 and hgc_cur eq 11))
    compute hgc_cur=-3
    compute enr_cur=-3 end if
compute tru=0
do repeat dip=dip1 to dip18
    if (dip gt 1) tru=1
    if (tru eq 1 and hgc_cur lt 12) hgc_cur=12
    if (tru eq 1 and (enr_cur eq 1 or enr_cur lt 0)) enr_cur=4
    if (hgc_cur eq -3 and enr_cur eq -4) enr_cur=-3
end repeat print
do if (flag00 eq 0) compute hgc_cur=-5
compute enr_cur=-5 end if
do if (hgc_cur eq -4 and flag00 eq 1) compute hgc_cur=-3
compute enr_cur=-3 end if
do if (enr_cur eq -4 and flag00 eq 1) compute enr_cur=-3

/* HGC_00=R(70071.) */ /* ENROLL_00=R(70072.) */
/* Each instance of -3 for highest grade completed and enrollment status is reviewed by checking selected school
variables. Values of -3 are recoded to valid values where auxilliary information indicates. However, there
are a number of -3s computed for highest grade completed and enrollment status that remain. */

```

NLSY79 APPENDIX 9:
LINKING EMPLOYERS
THROUGH SURVEY YEARS

LINKING EMPLOYERS THROUGH SURVEY YEARS

Two sets of variables from the main NLSY79 data set are necessary to link the employers from one contiguous year to the next. Comparable variables exist for all years allowing a link to be established through all contiguous interview years during which the respondent reported working for a specific employer. This document uses 1989 and 1990 for the purpose of demonstration. Both backward and forward employer linking is possible. The procedure for linking one employer or all employers through years is similar, and expands with the number of employers and years involved. Users should note that with the advent of the CAPI-administered surveys, beginning in 1993, the procedure has been simplified slightly. Beginning in 1993, only one variable per employer instead of two exists which may reflect the employer number from the previous survey year, if applicable. In addition, the CAPI instrument ensures that the CPS employer, if any, will always be employer #1. Therefore, it is only necessary to check the appropriate variable in the first employer supplement to determine CPS status. Although the same basic procedure must be followed to link employers through years, the number of variables that must be accessed is reduced.

BACKWARD LINKING OF EMPLOYERS

A first set of variables identifies the CPS employer from the most recent year (1990).

- 1) To Identify CPS Employer in 1990 (Work History Acronyms in Parentheses):

R 33400.00 (CPSJO901) Int Check 90 - Is Job # 1 Same As Current Job?
R 33540.00 (CPSJO902) Int Check 90 - Is Job # 2 Same As Current Job?
R 33680.00 (CPSJO903) Int Check 90 - Is Job # 3 Same As Current Job?
R 33820.00 (CPSJO904) Int Check 90 - Is Job # 4 Same As Current Job?
R 33960.00 (CPSJO905) Int Check 90 - Is Job # 5 Same As Current Job?

A “YES” answer (code 1) to one of these questions indicates that that employer was identified in the CPS section of the main questionnaire as the “CURRENT OR MOST RECENT” (CPS) employer and some basic information about that employer was collected in the CPS section for 1990. There should never be more than one CPS employer for any respondent in a given survey year. In **most** cases, the CPS employer will be JOB #1, **but not** in all cases. Therefore, all employers (1-5) must be checked.

Once the 1990 CPS employer has been identified, the next step is to determine if the CPS employer is a continuation of an employer reported in the previous contiguous interview year (1989 in this case). A second set of variables is used to identify which employer is the same as the 1990 CPS employer (or other employer if applicable) in the previous interview year (1989 in this case).

These variables are:

- 2) To Match or Link Any 1990 Employer with Its 1989 Employer Info (If Any) (Work History Acronyms in Parentheses):

R 33307.00 (PREV901) Employer Number From Info Sheet, Item 6 Matches, Job #1
R 33313.00 (PREV901) Employer Number From Info Sheet, Item 7 Matches, Job #1

R 33447.00 (PREV902) Employer Number From Info Sheet, Item 6 Matches, Job #2
R 33453.00 (PREV902) Employer Number From Info Sheet, Item 7 Matches, Job #2

R 33587.00 (PREV903) Employer Number From Info Sheet, Item 6 Matches, Job #3

R 33593.00 (PREV903) Employer Number From Info Sheet, Item 7 Matches, Job #3

R 33727.00 (PREV904) Employer Number From Info Sheet, Item 6 Matches, Job #4

R 33733.00 (PREV904) Employer Number From Info Sheet, Item 7 Matches, Job #4

R 33867.00 (PREV905) Employer Number From Info Sheet, Item 6 Matches, Job #5

R 33873.00 (PREV905) Employer Number From Info Sheet, Item 7 Matches, Job #5

This match or link is made by looking at two variables. (Each job has two comparable variables associated with it). These variables identify the number that a job was assigned in the previous interview year, if that job was reported. (INFO SHEET ITEM 6 lists the employers for whom the respondent was actually working at the date of the last interview. INFO SHEET ITEM 7 lists the employers the respondent mentioned during the previous interview but was not actually working for on the date of the previous interview.) If both of these variables contain a valid missing code (-4), then this job was not reported in the previous interview year (1989) and therefore can not be linked to any employer in the previous year. This is essentially a "new" employer, reported for the first time during the current survey year (or possibly an employer reported before the previous interview). If either of these variables (but not both) contains a valid number (1 or greater), this is the number of that job in the previous interview year. For example, if R(33400.) contains a "YES" (1) answer, then JOB #1 in 1990 has been identified as the CPS employer. One would then look at R(33307.) and R(33313.) to see if either of these variables contains a valid code. If R(33307.) contains a "2" for example, this would mean that the CPS employer for 1990 (JOB #1) is the same employer as JOB #2 from 1989. One could then attach information from JOB #1 in 1990 to information from the second employer (JOB #2) in 1989 as a continuing record of the respondent's experience with that employer. One could also check to see if this employer was the CPS employer in 1989 as well, by examining the 1989 variables which are comparable to the first set listed above. Using corresponding variables through contiguous interview years, information for an employer can be traced back through the first time the employer was ever reported by the respondent.

FORWARD LINKING

Forward linking can be accomplished in much the same way, using the same sets of corresponding variables listed above.

For example to link the 1989 CPS employer with its continuation (if any) in 1990, one would identify the 1989 CPS employer by looking at the set of 1989 variables comparable to the first set listed above. Once the 1989 CPS employer is identified, one would examine, the second set of variables for 1990 employers (the same as those listed above in this case), to determine if any of these jobs lists the number for the 1989 CPS employer as a "previous job". If one of these 1990 employer match variables contains the employer number of the 1989 CPS job, the match has been established. Information about the employer identified in 1990 could be attached to information about the 1989 CPS employer as a continuing record of the respondent's experience with that employer. For example, if employer number 1 (JOB #1) is identified as the 1989 CPS employer, one would need to search through the second set of variables listed above to see if any of them contained a "1" for previous employer. If one of the employer match variables for JOB #3 for instance, contains a "1", this would mean that JOB #3 in 1990 is the same as the CPS employer (JOB #1) in 1989.

This procedure works through **contiguous** survey years. For example, if a respondent was interviewed in all years from 1985-1990, a direct match can **not** be made between an employer reported in 1990 and one reported in 1985 without first establishing matches (or the lack thereof) through all intervening years. However if a respondent was interviewed in 1985, and not again until 1990, a link between employers

reported in these two years would be accomplished in the same manner as that described above between 1989 and 1990, as there would be no intervening year(s) to interfere with a direct match.

User Notes: Users should be aware of several characteristics of the data, described in the following paragraphs.

First, the NLSY79 employment history data are employer-based. All references to a “job” should be understood to be a reference to an employer. Information about work duties and positions and/or changes in duties or position performed or held during the respondent’s tenure with a specific employer is collected as part of the record for that specific employer. For example, a respondent may regard him/herself as having held a number of “jobs” or positions with employer #1 (JOB #1). However, any information collected about these different positions would all be regarded as information about employer #1 (JOB #1).

In survey years 1979–92, while the CPS employer is most often the first job listed in a given year, this is not always the case. Therefore, identification of the CPS employer (a common focus among researchers) requires that the “CPS employer identification” variables for each job be examined. Likewise, it is not always the case that a CPS employer in one year that has been reported before or since will be the CPS employer in other years. Likewise, a job which was not the CPS employer in the first year it was reported, may become the CPS employer in a later survey year. In survey years 1993–2000, the job order is determined by computer. The CPS employer for these survey years, if one is reported, is always the first employer.

Beginning in 1994, there is no specific variable identifying the first job explicitly as the CPS employer. The only exceptions to the rule that JOB #1 is the CPS job are respondents currently serving in the active military services, who have no CPS employer. The following equations identify these active military respondents, for whom employer #1 is **not** the CPS employer, in survey years 1994–2000.

$$\begin{aligned} & (R4536000 = 1 \text{ and } R4575300 > 0) \text{ (1994 active military respondents for whom job #1 is} \\ & \quad \text{not a CPS employer)} \\ & (R5231400 = 1 \text{ and } R5259400 > 0) \text{ (1996 active military respondents for whom job #1 is} \\ & \quad \text{not a CPS employer)} \\ & (R5830000 = 1 \text{ and } R5867400 > 0) \text{ (1998 active military respondents for whom job #1 is} \\ & \quad \text{not a CPS employer)} \\ & ((R6548700 = 1 \text{ or } R6549000 = 1) \text{ and } R6552000 > 0) \text{ (2000 active military respondents} \\ & \quad \text{for whom job #1 is not a CPS employer)} \end{aligned}$$

Dummy variables, comparable to the CPS job identification variables discussed above, can be created for 1994–2000 using these equations.

For survey years 1979–1992, the data file contains two possible variables to accesssed in identifying previous employer number of a given job. In survey years 1993–2000, the previous employer rostering is computerized. For those years, the NLSY79 data file contains one variable identifying previous employer number.

Beginning with the 2000 NLSY79 data release, the Work History data are included with the main data in a single data file. Researchers interested in working with employment history data should see “Appendix 18: Work History Data” in this document for further information.

NLSY79 APPENDIX 11:
NLSY79 ROUND 12 (1990) SURVEY
ADMINISTRATION METHODS

NLSY79 ROUND 12 (1990) SURVEY ADMINISTRATION METHODS

Round 12 of the NLSY79 survey incorporated a large-scale experiment involving comparison between PAPI (Paper-”Assisted” or Administered Personal Interviewing) and CAPI (Computer-Assisted Personal Interviewing) methods of interviewing, in anticipation of a possible conversion to CAPI-only data collection in the future. An experimental control-group design was implemented during the fielding period for the 1990 survey, allowing the examination of possible mode effects, any differences in data quality between modes, as well as differences in time and cost factors of administration. The CAPI version of the instrument was designed to replicate the paper instrument, which reduced the efficiency of the CAPI instrument somewhat. More detailed information on the results of examinations of mode effects are available through the Center. There are no indications that data quality was adversely affected by the experimental CAPI administration; indeed, the efficiency of data collection and data quality appear to have improved. Variables depicting interview modes in the 1990 data reflect respondents’ assignments to various design groups at the outset of the survey as well as the mode actually used, allowing researchers to conduct their own methodologically oriented examinations (see vars. R34003, R34004).

NLSY79 APPENDIX 12:
MOST IMPORTANT JOB LEARNING
ACTIVITIES (1993-94)

MOST IMPORTANT JOB LEARNING ACTIVITIES (1993)

Respondents were asked to identify the activities most important in helping them to learn how to perform their current or most recent job duties, and in helping them to learn about how work place changes would affect their jobs. This data is contained in the following variables:

- R41986. Most Important Activity To Learn Current/Most Recent Occupation (Re)
- R41987. Prompt-Most Important Ac To Learn Current/Most Recent Occupation (Re)
- R42641. Most Important Act To Learn How Wrkplc Chngs Would Affect Job
- R42642. Prompt-Most Important Act To Learn How Wrkplc Chngs Would Affect Job?

These variables contain only codes. The substantive value labels for the codes are listed below:

- 1 More than one activity mentioned as most important
- 2 Classes or seminars
- 3 Spending time with supervisors
- 4 Spending time with coworkers
- 5 Using self-teaching materials
- 6 Learning new skills on own
- 7 Trial and error
- 8 Previous job experience
- 9 On-the-job/hands on experience, learning by doing
- 10 Other
- 20 Reported only classes or seminars
- 30 Reported only spending time with supervisors
- 40 Reported only spending time with supervisors
- 50 Reported only self-teaching materials
- 60 Reported only learning new skills on own

MOST IMPORTANT JOB LEARNING ACTIVITIES (1994)

Respondents were asked to identify the activities most important in helping them to learn how to perform their current or most recent job duties, and in helping them to learn about how work place changes would affect their jobs. These data are contained in the following variables:

- R45931. Most Important Activity To Learn Job #1 Occupation (Re)
- R45932. Prompt-Most Important Ac To Learn Job #1 Occupation (Re)
- R46372. Most Important Activity To Learn Job #2 Occupation (Re)
- R46373. Prompt-Most Important Ac To Learn Job #2 Occupation (Re)
- R46809. Most Important Activity To Learn Job #3 Occupation (Re)
- R46810. Prompt-Most Important Ac To Learn Job #3 Occupation (Re)
- R47195. Most Important Activity To Learn Job #4 Occupation (Re)
- R47196. Prompt-Most Important Ac To Learn Job #4 Occupation (Re)
- R47530. Most Important Activity To Learn Job #5 Occupation (Re)
- R47531. Prompt-Most Important Ac To Learn Job #5 Occupation (Re)
- R48046. Most Important Act To Learn How Wrkplc Chngs Would Affect Job
- R48047. Prompt-Most Important Act To Learn How Wrkplc Chngs Would Affect Job?

These variables contain only codes. The substantive value labels for the codes are listed below:

- 1 More than one activity mentioned as most important
- 2 Classes or seminars
- 3 Spending time with supervisors
- 4 Spending time with coworkers
- 5 Using self-teaching materials
- 6 Learning new skills on own
- 7 Trial and error
- 8 Previous job experience
- 9 On-the-job/hands on experience, learning by doing
- 10 Other
- 11 Classes/workshops/conferences
- 12 Practicing
- 13 Read journals/books/articles/viewed videos
- 14 Communicate with/observe experts/colleagues/peers
- 15 Self-taught/studying on own
- 16 Experience (unspecified)
- 20 Reported only classes or seminars
- 30 Reported only spending time with supervisors
- 40 Reported only spending time with supervisors
- 50 Reported only self-teaching materials
- 60 Reported only learning new skills on own

NLSY79 APPENDIX 13

INTRODUCTION TO 1993–2000 CAPI

QUESTIONNAIRES AND CODEBOOKS

Introduction to 1993 CAPI Questionnaire and Codebook

Round 15 (1993) marks the first round of the National Longitudinal Survey of Youth to be administered entirely in CAPI (Computer-Assisted Personal Interviewing) mode. Wherever possible, comparability has been maintained between the 1993 data documentation and that of previous survey years. However, changes in technology and certain data collection procedures have resulted in some important differences. A number of these changes are discussed below.

I. NEW DOCUMENTATION ITEMS AND CHANGES IN EXISTING DOCUMENTATION

Questionnaire

An introduction to the 1993 NLSY79 questionnaire assists the user in understanding the format and content of the CAPI questionnaire. This document precedes the questionnaire.

Glossary of Save Array Names

A glossary of save array names accompanies the questionnaire. See Section II below and the 1993 NLSY79 Questionnaire introduction for a discussion of save arrays. These names are attached to data storage places which help the CAPI instrument to function. The save array names appear in numerous question texts in both the codebook and the questionnaire. The glossary of save array names defines the information stored in each save array as it is used during the course of the survey.

1993 Instrument Rosters

A listing of rosters, such as the Household Roster, has been added to this *Codebook Supplement* as Appendix 14. This listing includes the contents of the rosters, the applicable save array names attached to the rosters that might be encountered in the codebook and/or the questionnaire, and the variable reference number assigned to each piece of data in the codebook. Further detail about the rosters are provided in Section II.

Question-by-Question Specification (QxQ)

The 1993 QxQ is different in both appearance and content from those distributed during earlier rounds. Previously, the QxQ resembled an actual questionnaire, with the question-by-question specifications for the interviewer included. In essence, the QxQs could be used as a questionnaire. However, the 1993 QxQ is a combination of the electronic “help screens” that were available to the interviewer for specific questions, and a more detailed hard copy outline of the sections of the questionnaire and question intent. The electronic “help screens” contain listings of the questions for which they were used during the interview. The 1993 QxQ should be used in conjunction with the codebook or the questionnaire rather than as an independent document.

II. NEW TERMS

Save Arrays and “Text Fills”

Save arrays are reserved fields into which data can be placed, stored and accessed throughout the questionnaire. Each save array field is assigned a distinct name, often with an index number appended when the same piece of information is stored for a set of subjects. For instance, the save array “lintdate” contains the date of last interview for the respondent. The save arrays “child.name(1)” through “child.name(9)” contain the names of the first through the ninth biological child of the respondent. Throughout the survey, the last interview date and/or the name of the respondent’s first child can be accessed by invoking the save array names “lintdate” and “child.name(1)” respectively.

Users will encounter these save array names both in the codebook and in the questionnaire. Sometimes data in a save array is accessed and used to govern a skip or perform a calculation. At other times, a save array name may be part of a question text. In this case, the content of the specific save array content becomes part of the text of the question, and is read as such. This is referred to as a “text substitution” or “text fill”. For instance, a question text reading, “How often does (child.name(1)) see (his/her) (mom/dad)?”, would appear to the interviewer with the appropriate name replacing the save array “child.name(1)”, the appropriate gender replacing the save array “his/her” and the appropriate parent (mother or father) replacing the save array “mom/dad”. Such alternative wordings for text substitutions have been automated to a great extent with the advent of CAPI. In previous PAPI (Paper and Pencil Interviewing) questionnaires, the interviewer was required to make these choices as s/he read the question. (See the 1993 NLSY79 questionnaire introduction for more discussion of save arrays and text fills.)

Loops

As in PAPI instruments, certain sequences of questions in the CAPI instrument are repeated a number of times. For instance, some sets of questions are repeated up to 20 times in the Fertility Section (Section 9) of the 1993 CAPI questionnaire, for each of up to 20 children. Similarly, a sequence of questions is repeated ten times in the On Jobs Section, (Section 5), allowing up to ten new employers to be reported. Question names that include a “.1,” “.2,” “.3,” etc. at the end, belong to these repeating sequences of questions. Each repetition of the sequence of questions is referred to as a “loop.” Taking the Fertility Section as an example, the sequence of questions asking about the first child is referred to as the first “loop.” The sequence asking about the second child is referred to as the second “loop,” and so on. While the codebook generally includes more than one loop in a series (because more than one may contain valid data), the questionnaire includes only the **first loop** in each set of loops. (See the 1993 NLSY79 questionnaire introduction for further information.)

Dummy Records

A “dummy record” is a type of record or variable containing text that functions either as an explanatory statement or a transition. This text is read to the respondent, or less frequently, serves as an instruction to the interviewer. No information is entered when a dummy record appears on the screen. These screens are presented in questionnaire and hard-copy codebook. These dummy records however, are **not included** on the CD-ROM. When a dummy record is encountered in the codebook, it is present only to represent the flow of the questionnaire and generally appears as text preceding a variable with valid data. Although these dummy records have been assigned reference numbers, users should not attempt to use them as data items. These variables are marked by the text “(DUMMY)” at the end of the variable title.

Rosters and Roster Edit Records

A “roster” is a listing of one or more items of information pertaining to a specific set of subjects, such as the biological children of the respondent, or members of the respondent’s household. For instance, the CHILD roster contains a variety of information pertaining to each biological child of the respondent, (e.g., name, gender, birthdate, etc.). By using the roster format, an inventory of information can be gathered, tagged to a specific subject, carried along through the interview and accessed when necessary. This format also allows for these items of information to be presented to the interviewer at any time.

A “roster edit” is a type of record or variable in which the interviewer is presented with a roster or listing of information pertaining to a given set of subjects. The roster is then used by the interviewer to either verify information concerning this list of subjects, or to choose one of the subjects from the list, as indicated by the respondent. For example, after the respondent has reported all biological children, the interviewer is presented with the entire CHILD roster so that the s/he can verify with the respondent, the children listed. The interviewer is also presented with a modified roster of biological child information from which to choose the child (or children) for whom corrective information is required. These variables are marked by the text “(RE)” at the end of the variable title in the codebook or by the text “ROSTER EDIT” in a note contained in the codeblock.

III. CHANGES IN CODING CONVENTIONS

Hard and Soft Range Restrictions

The “Hard Minimum,” “Hard Maximum,” “Soft Minimum,” and “Soft Maximum” specifications control the allowable range of values that can be entered for a given question. These fields are only active when a question calls for the entry of a time date or amount. Questions that require the interviewer to select one response or to select all that apply do not contain this field, as the range limits are implicit in the distribution code block and are thereby enforced.

Hard minima and maxima are absolute limits that an interviewer- or respondent-generated answer must obey. Entry of values outside the hard range is not allowed. In such cases the interviewer is instructed to enter the maximum or minimum allowable value, as appropriate, enter the actual response in the comment field, and “flag” the case for central office checking. Soft minima and maxima are nested within the hard range. When a response falls outside the soft range but inside the hard range, the computer beeps and asks the interviewer to either confirm or change the response. These range checks are analogous to what took place in the central office coding shops, except that outliers are immediately brought to the interviewer’s attention so corrective action can be taken as necessary while the respondent is present. As one would expect, the number of callbacks to respondents to collect missing or seemingly incorrect data has fallen to virtually zero.

In some cases, the hard ranges are themselves determined by a variable. For example, event history data collection imposes certain logical limits on dates that are acceptable in subsequent questions. Whenever a range restriction is contingent on data collected during the current, or a previous, interview, this is indicated in the hard minimum or maximum field. When a question requires that the latest date that can be entered is the current interview date, the Hard Maximum field will contain the notation: Hard Max: Month[%curdate%] Day[%curdate%] Year[%curdate%]. This indicates that the various fields of the date are restricted so that no date later than that date can be entered. This is a powerful tool for event histories (among other types of data sequences) and is used extensively in the instrument. (See also the glossary of save array names and the 1993 NLSY79 questionnaire introduction for more information.)

Family Unit on Household Enumeration

In each survey year, a “family unit” code has been assigned to each member of the respondent’s household. This code identified whether the household member belonged to the respondent’s family unit or to other family units, possibly with other household members. Beginning with the 1993 survey, this family unit code has been collapsed into three codes. A code of “1” retains its traditional interpretation—a blood, marriage or adoptive relationship to the respondent. A new code of “2” represents a same sex or opposite sex partner and any household members related to the partner by blood or marriage. A code of “9” represents all other household members, with no familial relationship to either the respondent or his/her partner. This change was made in part because of the questionable reliability of these variables in past years, specifically as they were assigned to household members not in the respondent’s family unit.

IV. CHANGES IN PLACEMENT OF DATA AND/OR CODEBOOK ORDER

Employer Supplements

The Employer Supplement data are now placed in the codebook in the actual order of administration, directly following the CPS section (Section 6, in the 1993 instrument). Previously, Employer Supplement data had been placed after the main body of the survey, the information sheet data, and the household enumeration data.

1980 CPS Occupation and Industry Codes

In the 1993 CAPI data, the 1980 Occupation and Industry codes are contained in the Employer Supplement instead of in the CPS section, as has traditionally been the case. These variables were coded and entered independently of the actual interview data but were placed within the structure of the interview data for release.

Employer ID From Previous Survey Year

Employer IDs from the previous survey year have been placed directly preceding the wage questions in the Employer Supplements instead of directly preceding the start dates at the beginning of the Employer Supplements. In earlier PAPI rounds, the previous employer ID number was an independent data item, entered into the Employer Supplement by the interviewer during the course of the interview. Thus, these ids were part of the flow of the instrument. In the CAPI instrument this id number simply carried through the instrument in association with the appropriate employer on the EMPLOYER roster.

V. CHANGES IN APPEARANCE OR PRESENTATION OF DATA

Some 1993 CAPI questions generate more than one variable. For example, some questions collect information about the date an event happened and generate three variables: month, day and year. Similarly, questions that ask the respondent to indicate which of several responses are appropriate, and to pick all responses that apply, can generate multiple variables. In this case, each possible outcome becomes a distinct variable and is coded as “selected,” “not selected,” or a “valid skip.” When a question record generates multiple variables, those variables have decimals in the reference numbers. These two types of questions (date-entry and code-all-that-apply) are discussed further below.

Date-Entry Questions

All date questions, whether full dates (month, day and year) or just month and year, are represented in the CAPI codebook with respectively, three or two variables. Each variable contains the same codeblock displaying the ranges and missing values for all elements of the date. A base reference number ending in “.00”, is assigned to the first variable in the set. The same base reference number, ending in “.01” and “.02” if necessary, is assigned to the other elements of the date. Thus, if the month is assigned a reference number of R41454. (Date Left Armed Forces Branch Serving in at Time of Last Int (Srvd Active):Mo), the day and year would be assigned a reference number of R41454.01 and R41454.02 respectively. If the date is only a month and year, such as R41453. (“Date Left Armed Forces Branch Serving in at Time of Last Int (Served Inactive):Mo), then the year would be assigned R41453.01.

Code-All-That-Apply Questions

The NLSY79 includes questions that allow respondents to give multiple responses or code all responses that apply. These “code-all-that-apply” questions have traditionally been presented in the codebook as separate variables, with codes representing one specific substantive answer per variable, or an “applies”/“does not apply” coding scheme for each possible response. In the CAPI codebook, as in past codebooks, each possible response constitutes its own variable. Unlike such codeblocks in previous years, frequencies for all possible responses are represented in each codeblock for each variable. However, each specific variable contains only the valid data for one specific possible response. Reference numbers are assigned in the same manner as described for date-entry questions. A base reference number is assigned to the first possible response, such as R41752. (Method of Seeking Employment Past 4 Wks (Unemployed) Cps Item:1), with a decimal value being appended to that base number for each following possible response. In this example these numbers range from R41752.01 to R41752.08 (possible responses range from doing nothing, to checking with public and/or private employment agencies, friends, checking in newspapers, etc.). Although the codeblock for R41752.01 represents the frequencies for all possible responses to the question, accessing the data for that specific variable will produce only the data for the response “Checked with Public Employment Agency.”

Time Stamp Variables

In order to simplify the questionnaire and the public use data file, we have dropped a number of elapsed machine time variables. This detailed timing data may be useful to researchers interested in response times and how much thought the respondent may have given to particular questions or sets of questions. If you need such data, please notify CHRR and we will provide additional information.

VI. CHANGES IN DATA COLLECTION PROCEDURES WHICH AFFECT FORM AND/OR APPEARANCE OF NLSY79 1993 DATA**Machine-Generated Check Items**

Many questions represented in the NLSY79 data are “interviewer check items”. These are items used by the interviewer, to help determine the flow or skip patterns to follow in the questionnaire. In the codebook, these check items have traditionally looked like other questions that were asked of the respondent. In other words, they were often posed in the form of a question, which the interviewer simply asks and answers her/himself. Such items are still present in the CAPI instrument and codebook. However, a great number of them are now “machine checks”. Previously reported information is checked by machine and computations that once were made by the interviewer are made automatically, causing the appropriate skip pattern to be executed without intervention by the interviewer. In an effort to maintain comparability with past data releases and to clarify the skip patterns present in the instrument, a large number of these machine checks appear in the codebook. However the texts of these checks are in machine language and equations. Wherever possible, documentation comparable to previous check items in the PAPI questionnaires has been used. In addition, comments have been included as a “translation” to clarify the purpose of each machine check. R41750. (Int Check - Is R Currently Looking for Work? CPS Item) is one such variable. The text of the machine check is “[Q6-1]=3;”, an equation that is evaluated by the machine. A comment reading “/* was R ‘looking for work’ last week (Q6-1 coded ‘3’) */” explains what the machine is checking at this question.

“Consolidated” Variables

An effort has been made in the 1979-1993 data release to maintain comparability with previous releases in terms of data presentation. Toward this end, some variables have been “consolidated” with other variables. This means that the answers to more than one variable or set of variables are contained in a single variable or set of variables. This has generally been done in cases where data in previous years was available from a single variable or set of variables, but is now collected in the CAPI instrument in more than one variable or set of variables.

In each case, the variables being consolidated are mutually exclusive with respect to substantive responses. In other words, if variable A, variable B, and variable C are consolidated with each other, respondents will have given a response to only one of these—either variable A, variable B, or variable C. This allows the user to access one variable or a smaller set of variables, as may have been possible in previous rounds, instead of having to access a number of different variables to get the same information previously available in one.

These consolidated sets of variables are noted in the codebook documentation in the following manner. Variables that contain the responses from other variables have notes in the codebook documentation indicating that the variable is “Consolidated with Q#-#” and that it “Includes Responses from Q#-#”. For instance, R41463.-R41463.02 (Q4-11B - Date Began Service in Branch of Most Recent Armed Forces Enlistment (Srvd Active)) contain notes indicating that this set of variables is “Consolidated with Q4-11A” and that each variable in the set “Includes Responses from Q4-11A”. Conversely, R41462. and R41462.01 (Q4-11A, Date Began Service in Branch of Most Recent Armed Forces Enlistment (Srvd Inact)) contain notes indicating that this set of variables has been “Consolidated with Q4-11B” and that “Responses Included in Q4-11B”. In this case, the month/day/year dates of enlistment in the active

armed services have been consolidated with the month/year dates of enlistment in the inactive armed services. While both sets of variables are present in the questionnaire, users need only access R41463.–R41463.02 to get data for both variables. This was done to maintain comparability with the previous PAPI data releases, in which users could get information for both groups of respondents by accessing only one set of variables reflecting enlistment dates for both active and inactive armed forces.

Consolidation spares users from having to access a larger number of variables and use each separately or combine the responses themselves.

Changed Patterns and Formats for Data Collection

CAPI has led to a change in the formats in which some pieces of data are collected. Military dates are one example. For those enlisting in, or departing from, inactive military service since last interview, only the month and year of separation is collected, not the day. Those in active military service are asked the full date of separation or enlistment, including the day. The PAPI data contained the month and year, and then the day for those serving in the active forces or on active duty. The CAPI data however, contains two sets of date variables - a month and a year for those in the inactive forces and a month, a day and a year for those in the active forces or on active duty. While the data collected is identical to that in the PAPI survey, the format and presentation is slightly different. These variables have been consolidated into one set of variables in order to help maintain comparability with the PAPI data from past rounds (see above).

Identification of CPS Employer (Current/Most Recent Employer)

The mechanism for identifying the CPS employer (i.e. the current or most recent employer), has changed substantially with the implementation of a CAPI instrument. In previous PAPI rounds, identification of the CPS employer was dependent on the respondent reporting and interviewer decision-making. However, innovations in the CAPI instrument have allowed the CPS employer to be automatically identified. This is accomplished by sorting the list of employers from most recent to least recent, by stop date. In the unusual event that multiple employers have the same stop date, respondents are asked about the employer for whom they worked the most hours (last week/during the last week they worked), and the interviewer chooses the appropriate employer from a roster of potential CPS employers. If a unique CPS employer is identified (only one employer has the most recent stop date), the interviewer is presented immediately with a roster of employers to verify for accuracy with the respondent. The CPS employer appears at the top of the roster. Because the presence of the name of a first employer indicates that the respondent has worked since the date of last interview and that a CPS employer exists, there is no need to explicitly ask the respondent or to have the interviewer review information and enter an answer. Instead the answers to these questions are machine generated, after checking for the presence of a first employer name.

This results in some differences in appearance for these variables, compared to earlier PAPI rounds. Noticeable examples include R41774. (Q6-44), R41818. (Q6-48) and R41819. (Q6-49).

Possible Change in Order of Reporting and/or Repetition of Children (in Verification of Children's Record Form)

Users of the fertility data are familiar with a series of questions asked in the even-numbered survey years since 1986 in which the information on the Biological and Non-Biological Children's Record Form (CRF) is verified (including name, birthdate, gender and other elements). In the PAPI survey, the interviewer would verify the information for each child and correct any information necessary for that child, marking the information changed in the appropriate column. Properly administered, these verifications would have been made for each child in sequence, from first listed on the CRF to last. The CAPI series that approximates this verification series is structured in the same way. There are however, several differences in the way information is entered and in the possible order in which children are verified.

The interviewer is presented with the complete roster of children (biological or non-biological). On the first line of this roster is a line reading “All (Other) Child Information Correct.” The interviewer proceeds down the list of children from first to last, verifying the information for each child. When all information is correct for all children listed, s/he chooses the first line reading “All (Other) Children Correct” as the answer to that question. Instead of a variable for each child indicating whether the information for that child is correct, there is simply one response chosen at the point that the respondent confirms that there are no (more) errors in the information for the children listed.

If the respondent does indicate an error in the information for one or more children, there is no longer a formal order in which the children’s information is corrected. For instance, the respondent may have three children listed. Of these there may be errors in the information for children #2 and #3. If the respondent identifies the error in child #3’s information first, the interviewer is likely to identify that child first and correct his/her information first. In such a case, the first child identified who is in need of corrected information would be child #3 and the second child identified would be child #2. It is also possible for the respondent to indicate one problem with a child’s information, which the interviewer then corrects, and then for the respondent to indicate a second problem with the same child’s information, (which the interviewer may correct in a separate loop). In other words, the same child may be identified as the first and second child etc. needing corrected information, with each correction representing a separate piece of information on the roster.

While such patterns are not common, users of this verification data should not automatically assume that the children identified in this series are in the same order as they would be listed on the rosters. Nor should the assumption be made that each loop in this verification series contains a different child.

The final corrected data for all reported children, including corrections made during the verification sequence, can be found in record type CRFBIO for 1993. Reference numbers range between R(44127.) and R(44162.) These variables are the traditional variables included on the *Biological Children’s Record Form*.

Children Included in Cyclical Series (Feeding, Child Care)

Unlike the roster verification described above and the pregnancy question series described below, other question sequences pertaining to the children of respondents (particularly female respondents) now include automatic verification of each child for question eligibility. The feeding series and the Child Care Section of the questionnaire (section 10), are two such series of questions. In previous PAPI surveys, only children who were eligible for these series were present in the data. The CAPI instrument however, checks the eligibility of each child on the roster, first through last, to be administered these questions.

This results in differences in the patterns of data from those found in past rounds. Most noticeable will be the fact that each loop of these series will contain information on the corresponding child from the biological child roster. The first loop of the feeding series will contain information on the first child on the roster, the third loop will contain information on the third child, etc. In previous rounds the first loop might contain information on the third child (if that was the first one eligible for the series), etc. In addition, each loop will not necessarily contain substantive information. Only loops for eligible children would contain substantive information.

As noted above, users who wish to attach information on different aspects of female respondents’ children to child-specific records should be careful to check the specific child id of children on whom information is collected. This will ensure that the appropriate information is attached to the correct children in any child-based file.

Possible Change in Order of Reporting Children in “Pregnancy” Sequence

Children for whom pre-natal and neo-natal information is required are identified by the interviewer in similar fashion to those needing corrections to roster information. A roster is presented to the interviewer from which s/he chooses the appropriate child to which the “pregnancy” series in the Fertility Section should pertain. The CAPI instrument allows for information to be collected on up to five pregnancies since the last expanded fertility survey. Theoretically, children about whom pregnancy information is required should be identified from oldest to youngest. However, it is possible for the interviewer to enter a younger child first, and then ask questions about an older child. The assumption cannot be made therefore, that the first child for whom pre-natal/neo-natal information is collected is necessarily the first child born since the date of the last expanded fertility interview, and so on. However, the frequency of children being reported out of order should be low, because the instances of female respondents having more than one pregnancy since the last expanded fertility interview is low.

Users attempting to attach information on different aspects of female respondents’ children to child-specific records, either independently, or using the Child data sets as a base, should be careful to check the specific child id of children on whom information is collected, in order to ensure that the appropriate information is attached to the correct children.

Institution of Event Histories for Program Recipiency and Changes in Weeks of Unemployment**Recipiency**

In previous PAPI survey rounds, the program recipiency sequences, (including unemployment compensation for the respondent and respondent’s spouse, AFDC, government food stamps and other welfare/public assistance), consisted of a discrete reporting of each source of recipiency in the months of the calendar year preceding the survey year. In the CAPI instrument, all program recipiency sequences have been changed to an event history format, beginning with last reported month of receipt, and continuing through the current interview month. Information on between five and six spells of recipiency from each source is accommodated by the instrument (depending upon whether the respondent was receiving in December of the year before the last interview, the last month possible for him/her to have reported receipt as of the 1992 interview). The respondent reports the month in which s/he began receiving for each spell, and the month in which s/he stopped receiving for each spell. The average amount received per week/month during each spell was also collected.

One change from the PAPI survey rounds has been in our ability to discern the number of weeks during which unemployment compensation was collected. In PAPI survey instruments, a discrete figure for “number of weeks received unemployment compensation” in the calendar year preceding the survey year was collected from the respondent. In the CAPI instrument however, the dates of unemployment compensation receipt were only collected as month and year in which the spell(s) began and ended.

This results in specific week numbers being unavailable for calculating the exact number of weeks during a given calendar year, that a respondent received unemployment compensation. Instead, the beginning month and ending month of a spell are used to determine the number of months in which unemployment was received. Then, the number of months received is multiplied by 4.3 (average number of weeks in a month). This number is used as the total number of weeks received unemployment in a given calendar year. The average **amount** received is an average of the amounts reported during each spell falling within the appropriate calendar year. These are the figures used for number of weeks received unemployment, and average amount received per week by the respondent and his/her spouse, when calculating “Total Net Family Income.”

The possibility of over-estimation of the number of weeks received unemployment compensation exists under this strategy, as it is not always the case that receipt of unemployment during a particular month equates to receipt of unemployment for all weeks during that month. Any overestimation of weeks would probably range from one to seven or eight weeks. These weeks could fall in the beginning and ending

months of a spell of unemployment receipt. For a respondent who received unemployment for only one week in the beginning month and one week during the ending month of a spell, the estimation of weeks in that spell could exceed the actual weeks received by approximately seven or eight weeks. CHRR does not have reason to believe that this potential over-estimation of weeks of unemployment receipt is a major source of distortion in the calculation of “Total Net Family Income.”

Number of Program Recipiency Spells Reported

Each respondent is allowed to report up to five new program recipiency spells for each type of program. If five is insufficient, the opportunity is provided to report the month and year of most recent receipt. Those reporting that they received in December of the year before the previous interview year (the most recent month in which they could have reported receiving before the 1993 interview) were able to report an ending date for this spell that was technically still open at the date of last interview. This actually constitutes a sixth spell, on which information is collected **prior to** the five new spells allowed since the beginning of the last interview year.

An attempt has been made to reflect these differences in the variable titles. However, users should pay close attention to the variable titles and the variable content when using these variables. The questionnaire is very helpful in this regard. For those receiving in December of the year before the last interview year, information on the continuation of that spell is generally labelled spell #1, for those having received in December. However, the first of the five new spells allowed is also labelled spell #1. This is actually spell #2 for those who had an “open” spell entering the 1993 interview.

Negative Numbers in Data

In the course of the CAPI interview, negative numbers may result from calculations and be used subsequently in the interview. Two of these variables are present in the data set. These are R(43343.) and R(43353.) (difference between amount of child support the respondent and/or the respondent's spouse actually received and were supposed to receive in 1992, respectively). In the course of the CAPI survey, some of the values contained in these variables are actually negative values. However in processing the data for release, the negative signs have been dropped from the values. In order to determine if the values of these variables were originally negative values, users can take one of two steps: 1) perform the calculations themselves using the following formulas, R(43340.)-R(43341.) and R(43350.)-R(43351.) (for respondent and spouse respectively); or 2) check the two questions to which respondents with negative numbers would have been skipped, R(43346.) and R(43356.) for respondent and spouse respectively. If R(43346.) or R(43356.) have valid responses, then R(43343.) and R(43353.) respectively originally contained negative numbers which were converted to positive numbers for the purpose of including the figure in the text of the subsequent questions.

Respondent Not Included on Household Roster

Information about the respondent has traditionally been collected during the household interview, comparable to the information collected on all other household members at that stage of the interview. This information includes: relationship to respondent, age, gender, highest grade completed, and whether the household member worked during the calendar year preceding the interview year. Because all of this information for the respondent is either available elsewhere in the respondent's record and is not subject to change (relationship, age, gender), or is updated during the course of the survey (highest grade completed, working in past calendar year), it was not collected during the 1993 household interview as was the case in past rounds. Therefore, the respondent is not present on the household roster. Information on all other members of the household is collected as usual. This does not affect the family size variable, which is still computed using the household enumeration or roster. The respondent has been accounted for in the computation of family size. Researchers computing independent family size measures and wishing to include the respondent should remember to initialize the variable to “1”.

Initial Versus Final Versions of Information Sheet Variables

During the course of running the CAPI survey, certain information about the respondent as of the most recent interview is employed. Often the respondent is given an opportunity to verify this existing information before the interviewer proceeds to collect updated information. Many of the information sheet variables (contained in the area of interest LASTINFO) have either the notation “(Initial)” or “(Final)” appended to the variable title. A variable characterized as “(Initial)” reflects the value of that piece of data from the previous interview BEFORE the respondent was given a chance to verify its accuracy. A variable characterized as “(Final)” reflects the value of that piece of data AFTER the respondent has had a chance to verify, and possibly change its value. While the majority of cases will contain the same values on both the initial and final versions of these data, there are also a number of cases where those values are different. These are the cases where the respondent is disputing the initial value and has corrected or amended the information from the previous interview.

Introduction to 1994 CAPI Questionnaire and Codebook

The 1994 questionnaire and codebook continues many of the conventions established and discussed in the first part of this appendix. Whenever possible, comparability has been maintained between the 1994 data and documentation and that of previous survey years. However, there have been some significant changes, additions and improvements between the 1993 and 1994 data releases. The following is a discussion of some of these significant differences.

I. NEW DOCUMENTATION ITEMS AND CHANGES IN EXISTING DOCUMENTATION

Questionnaire

The 1994 printed questionnaire has been simplified. The essential elements for reading and following the flow of the questionnaire have been distilled. As a result, the 1994 questionnaire resembles much more closely the paper-and-pencil instruments that accompanied the 1979–92 data releases. See the 1994 questionnaire and accompanying documents for further discussion of format and contents.

New Appendices

Two new appendices have been added to this *Codebook Supplement*. “Appendix 16: The 1994 Recall Experiment” discusses an experiment that was conducted with the 1994 respondents to gain a better understanding of the effects of a biennial survey on respondent recall of life events. The second, “Appendix 15: Program Recipiency,” seeks to clarify the question modules devoted to program recipiency event histories (receipt of unemployment benefits, AFDC, Food Stamps and other welfare benefits), first implemented in the 1993 NLSY79 survey and continued in the 1994 wave.

II. CHANGES IN CODING CONVENTIONS

Class of Worker Variables

In 1994, the coding of the class of worker variables for each employer changed for the first time in the 16 waves of the NLSY79. This change resulted from the emulation of the substantially revised CAPI version of the actual Current Population Survey, in several modules of the NLSY79 (see discussion below). A category for “non-profit organization (including charitable)” was added. The response categories prior to 1994 were as follows:

- 1 An employee of a private company
- 2 A government employee
- 3 Self-employed in own business
- 4 Working without pay in a family business or farm

The response categories implemented in 1994 are as follows:

- 1 Government
- 2 Private for profit company
- 3 Non-profit organization (including charitable)
- 4 Self-employed
- 5 Working in a family business

Users may recode class of worker variables from previous years and/or create composite variables to achieve relative comparability in this set of variables between the 1979–93 variables and the 1994 variables (and those in future years). See the discussion below for further information on the manner in which class of worker questions are administered in the 1994 survey.

Employment Status Recode (ESR) Variables

Beginning in 1994, the substantive meaning of some of the codes assigned to the Employment Status Recode variables has changed and/or new substantive categories have been added. This was necessitated by the revision of the CPS section, modeled on the actual Current Population Survey (see discussion below). The Employment Status Recode variables were in turn modified to emulate the “Monthly Labor Status” variable computed by the Census Bureau from data generated from the Current Population Survey. Response codes prior to the 1994 were as follows:

<i>ESR</i>	<i>ESR (collapsed)</i>
1 Working	1 Employed
2 With a job not at work	2 Unemployed
3 Unemployed	3 Out of the labor force
4 Keeping house	4 In active forces
5 Going to school	
6 Unable to work	
7 Other	
8 In active forces	

The response categories implemented in 1994 are as follows:

<i>ESR</i>
1 Employed
2 Employed - absent from job
3 Unemployed - on layoff
4 Unemployed - looking for work
5 Not in labor force - retired
6 Disabled
7 Not in labor force - other

The categories for the collapsed version of ESR in 1994 are the same as in previous years.

Revised 1984-1994 FICE Code and 1979-1993 Highest Grade Completed and Enrollment Status Variables

Revised versions of the created Highest Grade Completed and Enrollment Status variables from 1979–94 have been added to the NLSY79 main data file. In addition, revised FICE code variables for 1984–94 have been added to the NLSY79 geocode data file. For each school identification (FICE code) variable, a variable containing a special edit code was added to the geocode data file as well. This data item identifies which of several possible types of codes were assigned to a given institution.

The original (unrevised) variables for Highest Grade Completed, Enrollment Status, and FICE code remain in the data files, along with the revised variables.

FICE Code Data

An examination by CHRR personnel of the NLSY79 1984–94 FICE code data resulted in the identification and correction of the following data problems:

- data entry error and other inaccuracies
- multiple FICE codes assigned to the same university
- institutions lacking FICE codes
- international universities
- unidentified institutions
- incorrectly identified invalid skips

Revisions to FICE code data was made in approximately 1250 cases. The FICE code data and accompanying special edit code variables are available only to those who satisfactorily complete the geocode licensing procedure. For further details concerning the revisions to the FICE code data, see “Attachment 105: Addendum to FICE CODES,” which is part of the *NLSY79 Geocode Codebook Supplement*.

Highest Grade Completed and Enrollment Status

Two main sources of error were identified in the Highest Grade Completed (HGC) variables from 1979–94:

- grade “reversals”, in which a respondent completed a lower grade in a later year, rather than staying in the same grade or advancing
- respondents with incomplete or ambiguous school information preventing the computation in a given year of a HGC or Enrollment Status (a problem which can then carry through subsequent years)

Examination of the longitudinal record resulted in one or more changes to the created Highest Grade Completed as of May 1st Survey Year variables from 1979–94 for approximately 3,500 respondents. Created enrollment status variables were revised where necessary, based on the revised HGC variables. See “Appendix 8: Highest Grade Completed and Enrollment Status Variable Creation” for details.

III. CHANGES IN PLACEMENT OF DATA AND/OR CODEBOOK ORDER

1980 CPS Occupation and Industry Codes

In the 1994 CAPI data, the 1980 Occupation and Industry codes for the CPS (current or most recent) employer are once again contained in the Employer Supplement. However unlike 1993, the variables for the 1980 Census codes are once again separate from those containing the 1970 occupation and industry codes for non-CPS employers in Employer Supplement # 1. See the discussion below for further information on the manner in which occupation and industry questions were administered in the 1994 survey.

Employer ID from Previous Survey Year

In the 1993 data file, the variables containing the employer IDs from the previous survey year were placed directly preceding the usual earnings questions in the appropriate Employer Supplement. In 1994, these variables have been returned to a position near the beginning of each employer supplement, similar to the 1979–92 survey years.

CPS and On Jobs Sections

In the NLSY79 1993 CAPI data file, the positions of the CPS and On Jobs (employer inventory) sections were changed from their traditional order in the 1979–92 waves. The 1993 On Jobs section was administered first, directly preceding the CPS section. This allowed the current/most recent (CPS) employer to be determined (a process which was automated in 1993) prior to administering the CPS section. The goal was to eliminate the error in collection of information on specific employers in the CPS section, and the variation in the order in which the Employer Supplements are administered.

Between 1979 and 1993, certain employer-specific questions pertaining to the CPS employer were contained in the CPS section itself, while similar or identical questions for all other employers were asked during the Employer Supplements. However, in 1994, all questions relating to specific employers were transferred to the Employer Supplements. This eliminated the need for the CPS employer to be established before the CPS section was administered. Therefore the order of the CPS and On Jobs sections have been shifted back to the 1979–92 pattern, with the CPS section being administered before the On Jobs section.

Class of Worker variables for new employers are collected in two questions, depending upon whether business ownership is reported by someone in the respondent's household.

IV. CHANGES IN DATA COLLECTION PROCEDURES WHICH AFFECT FORM AND/OR APPEARANCE OF NLSY79 1994 DATA

NLSY79 Emulation of the Revised CAPI Current Population Survey

Several segments of the 1994 survey were modified significantly in an effort to emulate as closely as possible related segments of the actual Current Population Survey. Specifically, these sections include the CPS section itself, questions on usual earnings with each employer and the segments which collect occupation, industry and class of worker for each employer.

CPS Section

The CPS section has been revised substantially. It continues to be modeled upon the Current Population Survey's section on activity in the last week and last four weeks, for which a conversion to a CAPI administration began in 1994. While essentially the same information is collected as in past years, it is somewhat more extensive and allows a more well-defined labor force status to be identified. In addition, specific information about the current or most recent employer (CPS employer) has been completely removed from the CPS section and placed in the first Employer Supplement. The current NLSY79 CPS section focuses only on general labor force status in specified time frames, not on any individual employer.

Usual Earnings

The questions concerning usual earnings of respondent and spouse/partner have also been revised based upon the Current Population Survey. In the NLSY79 1979-1993 surveys this information was collected in two questions, one for the actual rate of pay and one for the time unit for that rate of pay. Beginning with the 1994 NLSY79 survey, this series of questions has been expanded significantly. The series now allows more specificity in handling a given time unit for the rate of pay and includes the following elements:

- ◆ For hourly rates of pay:
 - Separate reporting of overtime pay if relevant and hours per week worked at the overtime rate
 - Reporting and/or verification of hours worked per week at the usual hourly rate of pay
 - Calculation of usual weekly wage taking overtime rate of pay and hours per week at both overtime and usual hourly rate of pay
 - Opportunity to correct specific data used in the calculation of usual weekly earnings and recalculate usual weekly earnings based upon corrected figures
- ◆ For annual rates of pay:
 - The number of weeks per year for which the respondent was paid
- ◆ For non-hourly rates of pay
 - Inclusion or non-inclusion of overtime amount in usual earning report for time period as appropriate, depending upon whether respondent said that s/he received any overtime pay
- ◆ For all rates of pay (hourly and non-hourly):
 - Bounding of figures reported for usual earnings, based either on the reported time period for the rate of pay or on the respondent's reports of usual earnings on previous interview dates with the same employer
 - Several opportunities for verification and/or correction of rates of pay that fall outside a certain range or that the respondent reports were recorded or calculated inaccurately

Occupation, Industry and Class of Worker

As with the usual earnings information and the CPS section itself, the series of questions soliciting information on occupation, industry and class of worker for each employer was revised to resemble those used in the actual Current Population Survey. The collection of this data was done in a manner relatively similar to that in the 1979-1993 surveys. The questions soliciting the descriptions of occupation, activities and duties and industry resembled those in past survey rounds. However, a pattern of verification of past information was adopted in the 1994 survey. If the employer is one reported during the previous interview, the description of the position given at the last interview of the occupation and activities and duties is read to the respondent. The respondent then confirms that the existing description is still correct, or says that it is not correct. If the existing description is reported to be incorrect, a new, updated and/or augmented description of the occupation and activities and duties is given. The industry and class of worker information is not recollected for pre-existing jobs, regardless of whether the occupation is changed or updated. If the respondent confirms that the description of the occupation is accurate from the previous interview, no new information is collected. The codes from the past interview for occupation, industry and class of worker are all retained in the 1994 data. For a newly reported employer, not present at the last interview, information is collected for all three types of data as appropriate, depending upon employer characteristics (number of hours worked per week, number of weeks worked with employer since the date of last interview/start date).

Self-Administered Drug Use Supplement

The 1994 NLSY79 CAPI instrument included a confidential self-administered Drug Use Supplement, resembling closely in content that included in the 1992 PAPI instrument. However, as with the rest of the survey, the Drug Supplement was administered as part of the electronic instrument, directly following the Income and Assets section of the questionnaire. When the interview reached the beginning of the Drug Use Supplement, the interviewer turned the laptop to the respondent and the respondent was asked to follow through the introductory instructional screens and then the actual Drug Use Supplement, choosing the responses him/herself.

With PAPI versions of the self-administered Drug Use Supplements, respondent confidentiality was maintained by supplying him/her with an envelope into which s/he would place the Drug Use Supplement after s/he was finished and seal. Field interviewers were not permitted to review the module. This data was connected to the respondent's entire record only after it reached the NORC central office staff and was data entered.

The CAPI version of the Drug Use Supplement employed an "electronic envelope" of sorts, in order to preserve the same level of confidentiality for the respondent. Once the respondent finished answering the questions and exited the range of questions within the Drug Use Supplement, that module was automatically hidden from view. This prevented interviewers in the field from reviewing the Drug Use Supplement and respondents' individual answers. The data could only be read once it was transmitted to the NORC central office and processed.

Introduction to 1996 CAPI Questionnaire and Codebook

I. CHANGES IN EXISTING DOCUMENTATION

Glossary of Save Array Names

The section entitled “Glossary of Save Array Names” in the 1993 and 1994 questionnaires has been eliminated in the 1996 Questionnaire. In the 1993 and 1994 questionnaires, the names of save arrays (data locations) appeared in the actual question text. Users could then look up the definitions for the data in the glossary of save array names. In 1996, the save array definitions have been inserted directly into the text of the questions in place of the save arrays. (See previous section titled “Introduction to 1993 CAPI Questionnaire and Codebook” in this appendix for further description of the term “save array.”)

II. CHANGES IN APPEARANCE OR PRESENTATION OF DATA

“Consolidated” Variables

The 1993 and 1994 data files contained “consolidated” variables. These were existing variables in which data from other variables had been combined, to produce one more inclusive data item. These variables intended to duplicate single data items as they existed in the paper-and-pencil data files prior to 1993. In the 1996 data file, no actual survey data points have been used to consolidate data from other survey variables. Instead, where appropriate, new variables have been created, and data from several questions consolidated into those created variables. These items include variables on marital status changes, payrates for the respondent and a spouse/partner if applicable, dates of military enlistment, and reasons for within-job gaps.

Recipiency History Variables

The 1979–96 NLSY79 release includes a large series of variables pertaining to the history of program recipiency for unemployment, AFDC, government food stamps, and SSI/other public assistance. Variables containing information on amounts received month-by-month from January 1978, and the source of data for each month, can be found in the record type RECIP_MON. Variables summarizing information on annual program receipt can be found in record type RECIP_YR.

The purpose of these variables is to provide users with a concentrated group of variables from which summary statistics on program receipt can more easily be constructed. For more information on this new series of variables, see “Appendix 15: Recipiency Event Histories.”

Introduction to 1998 CAPI Questionnaire and Codebook

I. CHANGES IN EXISTING DOCUMENTATION

Area of Interest

In previous rounds, the field Record Type on the data CD contained the basic topic of the data points. In this round, the title “Area of Interest” is now used in the same fashion and replaces the field Record Type.

II. CHANGES IN APPEARANCE OR PRESENTATION OF DATA

Introduction to the Use of the Tilda (~) in the Question Names

In previous rounds, question names were separated from additional information, such loop numbers, with an underscore. Beginning in 1998, the question names are separated from that other information with a tilda (~). The question types affected are discussed below.

Presentation of Dates in Question Names

Questions which contain date data are broken down into month, day, and year entries in the codebook. The question name is separated with a tilda, and the time unit is represented with an M, D, or Y. For example:

What month and year was that first diagnosed?

R63562.00	H40-CHRC-1A-M	This variable contains the month
R63562.01	H40-CHRC-1A-Y	This variable contains the year

Loops

In previous CAPI rounds, the fact that a question was part of a loop was indicated with one decimal and a loop number after the question name. In 1998, the convention has changed slightly with loop numbers now being indicated with two decimals and the loop number. For example:

Could you have returned to work last week if you had been recalled?

R58467.00	Q5-51.01	This variable contains loop number one.
R58468.00	Q5-51.02	This variable contains loop number two.

Select All Questions / Multiple Fields

Several questions allow the respondent to select many answers from an answer list or enter several pieces of information. In previous CAPI years, these answers were displayed with an underscore separating each answer choice based on the sequence. In 1998, each answer choice receives its own entry in the codebook, with a “1/0” value reflecting whether the answer choice in the pick list was chosen. For example:

Types of compensation based on performance

R60566.00	QES-PAYMT60A.01~000001	Answer w/ value 1 in loop 1
R60566.01	QES-PAYMT60A.01~000002	Answer w/ value 1 in loop 1
R60566.02	QES-PAYMT60A.01~000003	Answer w/ value 1 in loop 1
R60566.03	QES-PAYMT60A.01~000004	Answer w/ value 1 in loop 1
R60566.04	QES-PAYMT60A.01~000005	Answer w/ value 1 in loop 1
R60566.05	QES-PAYMT60A.01~000006	Answer w/ value 1 in loop 1

Introduction to 2000 CAPI Questionnaire and Codebook

The conventions found in the presentation of the 2000 questionnaire and codebook are very similar to those for the 1998 release. Several notable changes have been implemented in questionnaire content and the 2000 data release.

CPS Module Dropped for 2000

The CPS section is not included in the 2000 survey. It has been designated as a periodically rotating module. Much of the information gathered in the CPS section is contained in other forms elsewhere in the questionnaire.

Work History Data

For the first time with the 2000 data release, the work history data have been combined with the main data. A number of new areas of interest have been defined to contain data specifically created by the work history programs, such as the week-by-week arrays. This combined data set eliminates the need for separate extractions and merging of data from different data files. More information can be found in “Appendix 18: Work History Data” in this document.

Health Module for Respondents of 40+ Years

The 2000 survey administers the health module to respondents who have reached the age of 40 since the 1998 survey. Data for this module for the 1998 and 2000 respondents are presented in separate sets of codebook pages for each year. Users must combine data from both years to produce a complete set of data for all respondents through the 2000 interview who have been administered questions in this module.

NLSY79 APPENDIX 14:

1993-2000 INSTRUMENT ROSTERS

1993-2000 INSTRUMENT ROSTERS

During the course of the CAPI survey, a number of rosters, or matrices of data, are constructed. These rosters contain one or more pieces of information on a given subject. Rosters are often presented to the interviewers as lists of information that are used to verify information, or from which one of the subjects on the roster is chosen as the answer to a survey question. For example, the EMPLOYER roster (a list of employers for whom the employer has worked since the date of last interview), is presented to the interviewer at the end of the ON JOBS section (section 5 in the 1993 questionnaire, section 6 in the 1994–2000 questionnaires), so that s/he can verify that the list of employers, and some specific information associated with each employer is accurate.

Many of the rosters used during the administration of the survey are not presented as consolidated blocks of data in the public release data. Often in these cases, the relevant information contained in these rosters is present in variables scattered throughout the public data. For instance, although the EMPLOYER roster is not presented as a consolidated block, or a roster per se, much of the information contained in the roster is present in other variables in the Employer Supplements.

Schematic representations of both types of rosters are included in the following pages. The names of the roster fields are listed for each roster. Reference numbers are listed for roster fields which are also variables in the data set. This schematic presentation includes only the rosters for which the roster elements are included as variables in the data file. The HHI roster replaces the *Household Enumeration* from previous years. The CHILD roster in 1993 and the BIOCHILD rosters in 1994–2000, replace the *Biological Children's Record Form* from previous years. The NBIOCHILD rosters in 1994–2000 replace the *Non-Biological Children's Record Form* from previous years. The EMPLOYER roster did not exist in a cohesive fashion prior to 1993. However, most of the elements on this roster are included in a systematic fashion in the Employer Supplements in the 1979–2000 data releases. (For a more detailed discussion of rosters and save array names, see the section titled “Introduction to 1993 CAPI Questionnaire and Codebook” in Appendix 13.)

HHI ROSTER (1993)
(Household Enumeration)

PREFIX = "hh." (e.g. hhi.fu(1))

Family Unit	Sex	Relationship to Youth	Age	Highest Grade Completed	HH Member Work for Pay in 1992?
.fu(1) R44037.	.gender(1) R44038.	.relcode(1) R44039.	.age(1) R44040.	.graddecode(1) R44041.	.work(1) R44042.
.fu(2) R44043.	.gender(2) R44044.	.relcode(2) R44045.	.age(2) R44046.	.graddecode(2) R44047.	.work(2) R44048.
.fu(3) R44049.	.gender(3) R44050.	.relcode(3) R44051.	.age(3) R44052.	.graddecode(3) R44053.	.work(3) R44054.
.fu(4) R44055	.gender(4) R44056.	.relcode(4) R44057.	.age(4) R44058.	.graddecode(4) R44059	.work(4) R44060.
.fu(5) R44061.	.gender(5) R44062.	.relcode(5) R44063.	.age(5) R44064.	.graddecode(5) R44065.	.work(5) R44066.
.fu(6) R44067.	.gender(6) R44068.	.relcode(6) R44069.	.age(6) R44070.	.graddecode(6) R44071.	.work(6) R44072.
.fu(7) R44073.	.gender(7) R44074.	.relcode(7) R44075.	.age(7) R44076.	.graddecode(7) R44077.	.work(7) R44078.
.fu(8) R44079.	.gender(8) R44080.	.relcode(8) R44081.	.age(8) R44082.	.graddecode(8) R44083.	.work(8) R44084.
.fu(9) R44085.	.gender(9) R44086.	.relcode(9) R44087.	.age(9) R44088.	.graddecode(9) R44089.	.work(9) R44090.
.fu(10) R44091.	.gender(10) R44092.	.relcode(10) R44093.	.age(10) R44094.	.graddecode(10) R44095.	.work(10) R44096.
.fu(11) R44097.	.gender(11) R44098.	.relcode(11) R44099.	.age(11) R44100.	.graddecode(11) R44101.	.work(11) R44102.
.fu(12) R44103.	.gender(12) R44104.	.relcode(12) R44105.	.age(12) R44106.	.graddecode(12) R44107.	.work(12) R44108.
.fu(13) R44109.	.gender(13) R44110.	.relcode(13) R44111.	.age(13) R44112.	.graddecode(13) R44113.	.work(13) R44114.
.fu(14) R44115.	.gender(14) R44116.	.relcode(14) R44117.	.age(14) R44118.	.graddecode(14) R44119.	.work(14) R44120.
.fu(15) R44121.	.gender(15) R44122.	.relcode(15) R44123.	.age(15) R44124.	.graddecode(15) R44125.	.work(15) R44126.

EMPLOYER ROSTER (1993)

(Employers Worked for Since Date of Last Interview)

PREFIX = "employer." (e.g. employer.id(1))

ID# from Previous Survey Year	Name	Flag Indicating R Currently Working for Employer	Date Last Stopped Working for Employer	Date First Started Working For Employer	Is Employer CPS Employer?	Consecutive Day # of Employer Stopdate Date?	Employer Reported at Last Interview
.id(1) R41988.	.name(1)	.currflag(1) R41879.	.stopdate(1) R44403. R44403.01 R44403.02	.startdate(1) R41878. R41878.01 R41878.02	.cpsflag(1) R41931.	.stopday(1)	.prevflag(1) R41988.
.id(2) R42066.	.name(2)	.currflag(2) R42008.	.stopdate(2) R42012. R42012.01 R42012.02	.startdate(2) R42007. R42007.01 R42007.02	.cpsflag(2) #	.stopday(2)	.prevflag(2) R42066.
.id(3) R42138.	.name(3)	.currflag(3) R42084.	.stopdate(3) R42088. R42088.01 R42088.02	.startdate(3) R42083. R42083.01 R42083.02	.cpsflag(3) #	.stopday(3)	.prevflag(3) R42138.
.id(4) R42214.	.name(4)	.currflag(4) R42154.	.stopdate(4) R42158. R42158.01 R42158.02	.startdate(4) R42153. R42153.01 R42153.02	.cpsflag(4) #	.stopday(4)	.prevflag(4) R42214.
.id(5) R42286	.name(5)	.currflag(5) R42230.	.stopdate(5) R42234. R42234.01 R42234.02	.startdate(5) R42229. R42229.01 R42229.01	.cpsflag(5) #	.stopday(5)	.prevflag(5) R42286.
.id(6)	.name(6)	.currflag(6)	.stopdate(6)	.startdate(6)	.cpsflag(6)	.stopday(6)	.prevflag(6)
.id(7)	.name(7)	.currflag(7)	.stopdate(7)	.startdate(7)	.cpsflag(7)	.stopday(7)	.prevflag(7)
.id(8)	.name(8)	.currflag(8)	.stopdate(8)	.startdate(8)	.cpsflag(8)	.stopday(8)	.prevflag(8)
.id(9)	.name(9)	.currflag(9)	.stopdate(9)	.startdate(9)	.cpsflag(9)	.stopday(9)	.prevflag(9)
.id(10)	.name(10)	.currflag(10)	.stopdate(10)	.startdate(10)	.cpsflag(10)	.stopday(10))	.prevflag(10))

#NOTE: The CPS employer, if any exists, is now established by machine as the first employer. Therefore, the need for a question in Employer Supplements 2-5 to determine if any of these employers is the CPS employer is no longer necessary. Accordingly, only the first "cpsflag(1)" is accompanied by a reference number in this representation.

CHILD ROSTER (1993)
(Biological Child CRF)
PREFIX = "child." (e.g. child.name(1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R44127.	.gender(1) R44128.	.bdate(1) R44129.	.bdate(1) R44129.02	.statcode(1) R44130.
.id(2) R44131.	.gender(2) R44132.	.bdate(2) R44133.	.bdate(2) R44133.02	.statcode(2) R44134.
.id(3) R44135.	.gender(3) R44136.	.bdate(3) R44137.	.bdate(3) R44137.02	.statcode(3) R44138.
.id(4) R44139.	.gender(4) R44140.	.bdate(4) R44141.	.bdate(4) R44141.02	.statcode(4) R44142.
.id(5) R44143.	.gender(5) R44144.	.bdate(5) R44145.	.bdate(5) R44145.02	.statcode(5) R44146.
.id(6) R44147.	.gender(6) R44148.	.bdate(6) R44149.	.bdate(6) R44149.02	.statcode(6) R44150.
.id(7) R44151.	.gender(7) R44152.	.bdate(7) R44153.	.bdate(7) R44153.02	.statcode(7) R44154.
.id(8) R44155.	.gender(8) R44156.	.bdate(8) R44157.	.bdate(8) R44157.02	.statcode(8) R44158.
.id(9) R44159.	.gender(9) R44160.	.bdate(9) R44161.	.bdate(9) R44161.02	.statcode(9) R44162.

HHI ROSTER (1994)
(Household Enumeration)
PREFIX = "hh." (e.g. hhi.fu(1))

Family Unit	Sex	Relationship to Youth	Age	Highest Grade Completed	HH Member Work for Pay in 1993?
.fu(1) R50707.	.gender(1) R50708.	.relcode(1) R50709.	.age(1) R50710.	.gradecode(1) R50711.	.work(1) R50712.
.fu(2) R50713.	.gender(2) R50714.	.relcode(2) R50715.	.age(2) R50716.	.gradecode(2) R50717.	.work(2) R50718.
.fu(3) R50719.	.gender(3) R50720.	.relcode(3) R50721.	.age(3) R50722.	.gradecode(3) R50723.	.work(3) R50724.
.fu(4) R50725.	.gender(4) R50726.	.relcode(4) R50727.	.age(4) R50728.	.gradecode(4) R50729.	.work(4) R50730.
.fu(5) R50731.	.gender(5) R50732.	.relcode(5) R50733.	.age(5) R50734.	.gradecode(5) R50735.	.work(5) R50736.
.fu(6) R50737.	.gender(6) R50738.	.relcode(6) R50739.	.age(6) R50740.	.gradecode(6) R50741.	.work(6) R50742.
.fu(7) R50743.	.gender(7) R50744.	.relcode(7) R50745.	.age(7) R50746.	.gradecode(7) R50747.	.work(7) R50748.
.fu(8) R50749.	.gender(8) R50750.	.relcode(8) R50751.	.age(8) R50752.	.gradecode(8) R50753.	.work(8) R50754.
.fu(9) R50755.	.gender(9) R50756.	.relcode(9) R50757.	.age(9) R50758.	.gradecode(9) R50757.	.work(9) R50758.
.fu(10) R50761.	.gender(10) R50762.	.relcode(10) R50763.	.age(10) R50764.	.gradecode(10) R50765.	.work(10) R50766.
.fu(11) R50767.	.gender(11) R50768.	.relcode(11) R50769.	.age(11) R50770.	.gradecode(11) R50771.	.work(11) R50772.
.fu(12) R50773.	.gender(12) R50774.	.relcode(12) R50775.	.age(12) R50776.	.gradecode(12) R50777.	.work(12) R50778.
.fu(13) R50779.	.gender(13) R50780.	.relcode(13) R50781.	.age(13) R50782.	.gradecode(13) R50783.	.work(13) R50784.

Appendix 14: Instrument Rosters

EMPLOYER ROSTER (1994)

(Employers Worked for Since Date of Last Interview)

PREFIX = "employer." (e.g. employer.id (1))

ID# from Previous Survey Year	Name	Flag Indicating R Currently Working for Employer	Date Last Stopped Working for Employer	Date First Started Working for Employer	Is Employer CPS Employer?
.id(1) R45754.	.name(1)	.currflag(1) R45761.	.stopdate(1) R45772. R45772.01 R45772.02	.startdate(1) R45760. R45760.01 R45760.02	.cpsflag(1) #
.id(2) R46208.	.name(2)	.currflag(2) R46215.	.stopdate(2) R46226. R46226.01 R46226.02	.startdate(2) R46214. R46214.01 R46214.02	.cpsflag(2) #
.id(3) R46640.	.name(3)	.currflag(3) R46647.	.stopdate(3) R46658. R46658.01 R46658.02	.startdate(3) R46646. R46646.01 R46646.02	.cpsflag(3) #
.id(4) R47057.	.name(4)	.currflag(4) R47063.	.stopdate(4) R47074. R47074.01 R47074.02	.startdate(4) R47062. R47062.01 R47062.02	.cpsflag(4) #
.id(5) R47405.	.name(5)	.currflag(5) R47411.	.stopdate(5) R47421. R47421.01 R47421.02	.startdate(5) R47410. R47410.01 R47410.02	.cpsflag(5) #
.id(6)	.name(6)	.currflag(6)	.stopdate(6)	.startdate(6)	.cpsflag(6)
.id(7)	.name(7)	.currflag(7)	.stopdate(7)	.startdate(7)	.cpsflag(7)
.id(8)	.name(8)	.currflag(8)	.stopdate(8)	.startdate(8)	.cpsflag(8)
.id(9)	.name(9)	.currflag(9)	.stopdate(9)	.startdate(9)	.cpsflag(9)
.id(10)	.name(10)	.currflag(10)	.stopdate(10)	.startdate(10)	.cpsflag(10)

#NOTE: The CPS employer, if any exists, is now established by machine as the first employer. Therefore, the need for a question in Employer Supplements 2-5 to determine if any of these employers is the CPS employer is no longer necessary. Accordingly, only the first "cpsflag(1)" is accompanied by a reference number in this representation.

BIOCHILD ROSTER (1994)

(**Biological Child CRF**)

PREFIX = "biochild." (e.g. biochild.id (1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R50608.	.gender(1) R50609.	.bdate(1) R50610.	.bdate(1) R50610.02	.statcode(1) R50612.
.id(2) R50616.	.gender(2) R50617.	.bdate(2) R50618.	.bdate(2) R50618.02	.statcode(2) R50620.
.id(3) R50624.	.gender(3) R50625.	.bdate(3) R50626.	.bdate(3) R50626.02	.statcode(3) R50628.
.id(4) R50632.	.gender(4) R50633.	.bdate(4) R50634.	.bdate(4) R50634.02	.statcode(4) R50636.
.id(5) R50640.	.gender(5) R50641.	.bdate(5) R50642.	.bdate(5) R50642.02	.statcode(5) R50644.
.id(6) R50648.	.gender(6) R50649.	.bdate(6) R50650.	.bdate(6) R50650.02	.statcode(6) R50652.
.id(7) R50656.	.gender(7) R50657.	.bdate(7) R50658.	.bdate(7) R50658.02	.statcode(7) R50660.
.id(8) R50664.	.gender(8) R50665.	.bdate(8) R50666.	.bdate(8) R50666.02	.statcode(8) R50668.

NBIOCHILD ROSTER (1994)

(**Non-biological Child CRF**)

PREFIX = "nbiochild." (e.g. nbiochild.id (1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R50672.	.gender(1) R50673.	.bdate(1) R50674.	.bdate(1) R50674.02	.statcode(1) R50675.
.id(2) R50677.	.gender(2) R50678.	.bdate(2) R50679.	.bdate(2) R50679.02	.statcode(2) R50680.
.id(3) R50682.	.gender(3) R50683.	.bdate(3) R50684.	.bdate(3) R50684.02	.statcode(3) R50685.
.id(4) R50687.	.gender(4) R50688.	.bdate(4) R50689.	.bdate(4) R50689.02	.statcode(4) R50690.
.id(5) R50692.	.gender(5) R50693.	.bdate(5) R50694.	.bdate(5) R50694.02	.statcode(5) R50695.
.id(6) R50697.	.gender(6) R50698.	.bdate(6) R50699.	.bdate(6) R50699.02	.statcode(6) R50700.
.id(7) R50702.	.gender(7) R50703.	.bdate(7) R50704.	.bdate(7) R50704.02	.statcode(7) R50705.

HHI ROSTER (1996)
(Household Enumeration)
 PREFIX = "hh." (e.g. hhi.fu(1))

Family Unit	Sex	Relationship to Youth	Age	Highest Grade Completed	HH Member Work for Pay in 1995?
.fu(1) R57486.	.gender(1) R57487.	.relcode(1) R57488.	.age(1) R57489.	.graddecode(1) R57490.	.work(1) R57491.
.fu(2) R57492.	.gender(2) R57493.	.relcode(2) R57494.	.age(2) R57495.	.graddecode(2) R57496.	.work(2) R57497.
.fu(3) R57498.	.gender(3) R57499.	.relcode(3) R57500.	.age(3) R57501.	.graddecode(3) R57502.	.work(3) R57503.
.fu(4) R57504.	.gender(4) R57505.	.relcode(4) R57506.	.age(4) R57507.	.graddecode(4) R57508.	.work(4) R57509.
.fu(5) R57510.	.gender(5) R57511.	.relcode(5) R57512.	.age(5) R57513.	.graddecode(5) R57514.	.work(5) R57515.
.fu(6) R57516.	.gender(6) R57517.	.relcode(6) R57518.	.age(6) R57519.	.graddecode(6) R57520.	.work(6) R57521.
.fu(7) R57522.	.gender(7) R57523.	.relcode(7) R57524.	.age(7) R57525.	.graddecode(7) R57526.	.work(7) R57527.
.fu(8) R57528.	.gender(8) R57529.	.relcode(8) R57530.	.age(8) R57531.	.graddecode(8) R57532.	.work(8) R57533.
.fu(9) R57534.	.gender(9) R57535.	.relcode(9) R57536.	.age(9) R57537.	.graddecode(9) R57538.	.work(9) R57539.
.fu(10) R57540.	.gender(10) R57541.	.relcode(10) R57542.	.age(10) R57543.	.graddecode(10) R57544.	.work(10) *
.fu(11) R57545.	.gender(11) R57546.	.relcode(11) R57547.	.age(11) R57548.	.graddecode(11) R57549.	.work(11) *
.fu(12) R57550.	.gender(12) R57551.	.relcode(12) R57552.	.age(12) *	.graddecode(12) *	.work(12) *

EMPLOYER ROSTER (1996)

(Employers Worked for Since Date of Last Interview)

PREFIX = "employer." (e.g. employer.id (1))

ID# from Previous Survey Year	Name	Flag Indicating R Currently Working for Employer	Date Last Stopped Working for Employer	Date First Started Working for Employer	Is Employer CPS Employer?
.id(1) R52596.	.name(1)	.currflag(1) R52603.	.stopdate(1) R52614. R52614.01 R52614.02	.startdate(1)+ R52601. R52602. R52601.01 R52602.01 R52601.02 R52602.02	.cpsflag(1) #
.id(2) R53005.	.name(2)	.currflag(2) R53012.	.stopdate(2) R53023. R53023.01 R53023.02	.startdate(2)+ R53010. R53011. R53010.01 R53011.01 R53010.02 R53011.02	.cpsflag(2) #
.id(3) R53400.	.name(3)	.currflag(3) R53407.	.stopdate(3) R53418. R53418.01 R53418.02	.startdate(3)+ R53405. R53406. R53405.01 R53406.01 R53405.02 R53406.02	.cpsflag(3) #
.id(4) R53778.	.name(4)	.currflag(4) R53784.	.stopdate(4) R53795. R53795.01 R53795.02	.startdate(4)+ R53782. R53783. R53782.01 R53783.01 R53782.02 R53783.02	.cpsflag(4) #
.id(5) R54127.	.name(5)	.currflag(5) R54134.	.stopdate(5) R54144. R54144.01 R54144.02	.startdate(5)+ R54132. R54133. R54132.01 R54133.01 R54132.02 R54133.02	.cpsflag(5) #
.id(6)	.name(6)	.currflag(6)	.stopdate(6)	.startdate(6)	.cpsflag(6)
.id(7)	.name(7)	.currflag(7)	.stopdate(7)	.startdate(7)	.cpsflag(7)
.id(8)	.name(8)	.currflag(8)	.stopdate(8)	.startdate(8)	.cpsflag(8)
.id(9)	.name(9)	.currflag(9)	.stopdate(9)	.startdate(9)	.cpsflag(9)
.id(10)	.name(10)	.currflag(10)	.stopdate(10)	.startdate(10)	.cpsflag(10)

#NOTE: The CPS employer, if any exists, is now established by machine as the first employer. Therefore, the need for a question in Employer Supplements 2-5 to determine if any of these employers is the CPS employer is no longer necessary. Accordingly, only the first "cpsflag(1)" is accompanied by a reference number in this representation.

+NOTE: Startdates for employers are contained in one of two sets of variables, depending upon whether the respondent was working for the employer at the date of last interview. Both sets are listed above. These two sets of variables combined, contain start dates for all respondents with each employer.

BIOCHILD ROSTER (1996)

(**Biological Child CRF**)

PREFIX = "biochild." (e.g. biochild.id (1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R57382.	.gender(1) R57390.	.bdate(1) R57383.	.bdate(1) R57383.02	.statcode(1) R57385.
.id(2) R57391.	.gender(2) R57399.	.bdate(2) R57392.	.bdate(2) R57392.02	.statcode(2) R57394.
.id(3) R57400.	.gender(3) R57408.	.bdate(3) R57401.	.bdate(3) R57401.02	.statcode(3) R57403.
.id(4) R57409.	.gender(4) R57416.	.bdate(4) R57410.	.bdate(4) R57410.02	.statcode(4) R57412.
.id(5) R57417.	.gender(5) R57424.	.bdate(5) R57418.	.bdate(5) R57418.02	.statcode(5) R57420.
.id(6) R57425.	.gender(6) R57431.	.bdate(6) R57426.	.bdate(6) R57426.02	.statcode(6) R57428.
.id(7) R57432.	.gender(7) R57438.	.bdate(7) R57433.	.bdate(7) R57433.02	.statcode(7) R57435.
.id(8) R57439.	.gender(8) R57445.	.bdate(8) R57440.	.bdate(8) R57440.02	.statcode(8) R57442.
.id(9) R57446.	.gender(9) R57451.	.bdate(9) R57447.	.bdate(9) R57447.02	.statcode(9) R57449.

NBIOCHILD ROSTER (1996)

(**Non-biological Child CRF**)

PREFIX = "nbiochild." (e.g. nbiochild.id (1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R57456.	.gender(1) R57452.01	.bdate(1) R57453.	.bdate(1) R57453.02	.statcode(1) R57454.
.id(2) R57461.	.gender(2) R57457.01	.bdate(2) R57458.	.bdate(2) R57458.02	.statcode(2) R57459.
.id(3) R57466.	.gender(3) R57462.01	.bdate(3) R57463.	.bdate(3) R57463.02	.statcode(3) R57464.
.id(4) R57471.	.gender(4) R57467.01	.bdate(4) R57468.	.bdate(4) R57468.02	.statcode(4) R57469.
.id(5) R57476.	.gender(5) R57472.01	.bdate(5) R57473.	.bdate(5) R57473.02	.statcode(5) R57474.
.id(6) R57481.	.gender(6) R57477.01	.bdate(6) R57478.	.bdate(6) R57478.02	.statcode(6) R57479.
.id(7) R57485.	.gender(7) R57482.01	.bdate(7) R57483.	.bdate(7) R57483.02	.statcode(7) R57483.10

HHI ROSTER (1998)
(Household Enumeration)
PREFIX = "hh." (e.g. hhi.fu(1))

Family Unit	Sex	Relationship to Youth	Age	Highest Grade Completed	HH Member Work for Pay in 1996?
.fu(1) R64491.	.gender(1) R64506.	.relcode(1) R64521.	.age(1) R64536.	.gradecode(1) R64551.	.work(1) R64566.
.fu(2) R64492.	.gender(2) R64507.	.relcode(2) R64522.	.age(2) R64537.	.gradecode(2) R64552.	.work(2) R64567.
.fu(3) R64493.	.gender(3) R64508.	.relcode(3) R64523.	.age(3) R64538.	.gradecode(3) R64553.	.work(3) R64568.
.fu(4) R64494.	.gender(4) R64509.	.relcode(4) R64524.	.age(4) R64539.	.gradecode(4) R64554.	.work(4) R64569.
.fu(5) R64495.	.gender(5) R64510.	.relcode(5) R64525.	.age(5) R64540.	.gradecode(5) R64555.	.work(5) R64570.
.fu(6) R64496.	.gender(6) R64511.	.relcode(6) R64526.	.age(6) R64541.	.gradecode(6) R64556.	.work(6) R64571.
.fu(7) R64497.	.gender(7) R64512.	.relcode(7) R64527.	.age(7) R64542.	.gradecode(7) R64557.	.work(7) R64572.
.fu(8) R64498.	.gender(8) R64513.	.relcode(8) R64528.	.age(8) R64543.	.gradecode(8) R64558.	.work(8) R64573.
.fu(9) R64499.	.gender(9) R64514.	.relcode(9) R64529.	.age(9) R64544.	.gradecode(9) R64559.	.work(9) R64574.
.fu(10) R64500.	.gender(10) R64515.	.relcode(10) R64530.	.age(10) R64545.	.gradecode(10) R64560.	.work(10) R64575.
.fu(11) R64501.	.gender(11) R64516.	.relcode(11) R64531.	.age(11) R64546.	.gradecode(11) R64561.	.work(11) R64576.
.fu(12) R64502.	.gender(12) R64517.	.relcode(12) R64532.	.age(12) R64547.	.gradecode(12) R64562.	.work(12) R64577.

Appendix 14: Instrument Rosters

EMPLOYER ROSTER (1998)

(Employers Worked for Since Date of Last Interview)

PREFIX = "employer." (e.g. employer.id (1))

ID# from Previous Survey Year	Name	Flag Indicating R Currently Working for Employer	Date Last Stopped Working for Employer	Date First Started Working for Employer	Is Employer CPS Employer?
.id(1) R64579.	.name(1)	.currflag(1) R65025.	.stopdate(1) R64591. R64591.01 R64591.02	.startdate(1) R64603. R64603.01 R64603.02	.cpsflag(1) R64615.
.id(2) R64580.	.name(2)	.currflag(2) R65026.	.stopdate(2) R64592. R64592.01 R64592.02	.startdate(2) R64604. R64604.01 R64604.02	.cpsflag(2) R64616.
.id(3) R64581.	.name(3)	.currflag(3) R65027.	.stopdate(3) R64593. R64593.01 R64593.02	.startdate(3) R64605. R64605.01 R64605.02	.cpsflag(3) R64617.
.id(4) R64582.	.name(4)	.currflag(4) R65028.	.stopdate(4) R64594. R64594.01 R64594.02	.startdate(4) R64606. R64606.01 R64606.02	.cpsflag(4) R64618.
.id(5) R64583.	.name(5)	.currflag(5) R65029.	.stopdate(5) R64595. R64595.01 R64595.02	.startdate(5) R64607. R64607.01 R64607.02	.cpsflag(5) R64619.
.id(6)	.name(6)	.currflag(6)	.stopdate(6)	.startdate(6)	.cpsflag(6)
.id(7)	.name(7)	.currflag(7)	.stopdate(7)	.startdate(7)	.cpsflag(7)
.id(8)	.name(8)	.currflag(8)	.stopdate(8)	.startdate(8)	.cpsflag(8)
.id(9)	.name(9)	.currflag(9)	.stopdate(9)	.startdate(9)	.cpsflag(9)
.id(10)	.name(10)	.currflag(10)	.stopdate(10)	.startdate(10)	.cpsflag(10)

BIOCHILD ROSTER (1998)

(**Biological Child CRF**)

PREFIX = "biochild4." (e.g. biochild4.id (1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R64615.	.gender(1) R64434.	.bdate(1) R64372.01	.bdate(1) R64372.02	.statcode(1) R64379.
.id(2) R64616.	.gender(2) R64435.	.bdate(2) R64373.01	.bdate(2) R64373.02	.statcode(2) R64380.
.id(3) R64617.	.gender(3) R64436.	.bdate(3) R64374.01	.bdate(3) R64374.02	.statcode(3) R64381.
.id(4) R64618.	.gender(4) R64437.	.bdate(4) R64375.01	.bdate(4) R64375.02	.statcode(4) R64382.
.id(5) R64619.	.gender(5) R64438.	.bdate(5) R64376.01	.bdate(5) R64376.02	.statcode(5) R64383.
.id(6) R64620.	.gender(6) R64434.	.bdate(6) R64377.01	.bdate(6) R64377.02	.statcode(6) R64384.
.id(7) R64621.	.gender(7) R64435.	.bdate(7) R64378.01	.bdate(7) R64378.02	.statcode(7) R64385.
.id(8) R64622.	.gender(8) R64436.	.bdate(8) R64379.01	.bdate(8) R64378.02	.statcode(8) R64386.
.id(9) R64623.	.gender(9) R64437.	.bdate(9) R64380.01	.bdate(9) R64380.02	.statcode(9) R64387.

NBIOCHILD ROSTER (1998)

(**Non-biological Child CRF**)

PREFIX = "nbiochild4." (e.g. nbiochild4.id (1))

ID Number	Sex	Month of Birth	Year of Birth	Status
.id(1) R64445.	.gender(1) *	.bdate(1) R64454..01	.bdate(1) R64454..02	.statcode(1) R64463.
.id(2) R64446.	.gender(2) *	.bdate(2) R64455.01	.bdate(2) R64455..02	.statcode(2) R64464.
.id(3) R64447.	.gender(3) *	.bdate(3) R64456.01	.bdate(3) R64456.02	.statcode(3) R64465.
.id(4) R64448.	.gender(4) *	.bdate(4) R64457.01	.bdate(4) R64457.02	.statcode(4) R64466.
.id(5) R64449.	.gender(5) *	.bdate(5) R64458.01	.bdate(5) R64458.02	.statcode(5) R64467.
.id(6) R64450.	.gender(6) *	.bdate(6) R64459.01	.bdate(6) R64459.02	.statcode(6) R64468.
.id(7) R64451.	.gender(7) *	.bdate(7) R64460.01	.bdate(7) R64460.02	.statcode(7) R64469.

HHI ROSTER (2000)
(Household Enumeration)
 PREFIX = "hh." (e.g. hhi.fu(1))

Family Unit	Sex	Relationship to Youth	Age	Highest Grade Completed	HH Member Work for Pay in 1999?
.fu(1)	.gender(1) R69881.	.relcode(1) R69915.	.age(1) R69932.	.graddecode(1) R69949.	.work(1) R69966.
.fu(2)	.gender(2) R69882.	.relcode(2) R69916.	.age(2) R69933.	.graddecode(2) R69950.	.work(2) R69967.
.fu(3)	.gender(3) R69883.	.relcode(3) R69917.	.age(3) R69934.	.graddecode(3) R69951.	.work(3) R69968.
.fu(4)	.gender(4) R69884.	.relcode(4) R69918.	.age(4) R69935.	.graddecode(4) R69952.	.work(4) R69969.
.fu(5)	.gender(5) R69885.	.relcode(5) R69919.	.age(5) R69936.	.graddecode(5) R69953.	.work(5) R69970.
.fu(6)	.gender(6) R69886.	.relcode(6) R69920.	.age(6) R69937.	.graddecode(6) R69954.	.work(6) R69971.
.fu(7)	.gender(7) R69887.	.relcode(7) R69921.	.age(7) R69938.	.graddecode(7) R69955.	.work(7) R69972.
.fu(8)	.gender(8) R69888.	.relcode(8) R69922.	.age(8) R69939.	.graddecode(8) R69956.	.work(8) R69973.
.fu(9)	.gender(9) R69889.	.relcode(9) R69923.	.age(9) R69940.	.graddecode(9) R69957.	.work(9) R69974.
.fu(10)	.gender(10) R69890.	.relcode(10) R69924.	.age(10) R69941.	.graddecode(10) R69958.	.work(10) R69975.
.fu(11)	.gender(11) R69891.	.relcode(11) R69925.	.age(11) R69942.	.graddecode(11) R69959.	.work(11) R69976.
.fu(12)	.gender(12) R69892.	.relcode(12) R69926.	.age(12) R69943.	.graddecode(12) R69960.	
.fu(13)	.gender(13) R69893.	.relcode(13) R69927.	.age(13) R69944.	.graddecode(13) R69961.	
.fu(14)	.gender(14) R69894.	.relcode(14) R69928.	.age(14) R69945.	.graddecode(14) R69962.	.work(14) R69977.
.fu(15)	.gender(15) R69895.	.relcode(15) R69929.	.age(15) R69946.	.graddecode(15) R69963.	
.fu(16)	.gender(16) R69896.	.relcode(16) R69930.	.age(16) R69947.	.graddecode(16) R69964.	
.fu(17)	.gender(17) R69897.	.relcode(17) R69931.	.age(17) R69948.	.graddecode(17) R69965.	.work(17) R69978.

EMPLOYER ROSTER (2000)

(Employers Worked for Since Date of Last Interview)

PREFIX = "employer." (e.g. employer.id (1))

ID# from Previous Survey Year	Name	Flag Indicating R Currently Working for Employer	Date Last Stopped Working for Employer	Date First Started Working for Employer	Is Employer CPS Employer?
.id(1) R65527.	.name(1)	.curflag(1) R65550.	.stopdate(1) R69999. (d) R69999.01 (m) R69999.02 (y)	.startdate(1) R70009. (d) R70009.01 (m) R70009.02 (y)	.cpsflag(1) R70019.
.id(2) R65528.	.name(2)	.curflag(2) R65551.	.stopdate(2) R70000. (d) R70000.01 (m) R70000.02 (y)	.startdate(2) R70010. (d) R70010.01 (m) R70010.02 (y)	.cpsflag(2) R70020.
.id(3) R65529.	.name(3)	.curflag(3) R65552.	.stopdate(3) R70001. (d) R70001.01 (m) R70001.02 (y)	.startdate(3) R70011. (d) R70011.01 (m) R70011.02 (y)	.cpsflag(3) R70021.
.id(4) R65530.	.name(4)	.curflag(4) R65553.	.stopdate(4) R70002. (d) R70002.01 (m) R70002.02 (y)	.startdate(4) R70012. (d) R70012.01 (m) R70012.02 (y)	.cpsflag(4) R70022.
.id(5) R65531.	.name(5)	.curflag(5) R65554.	.stopdate(5) R70003. (d) R70003.01 (m) R70003.02 (y)	.startdate(5) R70013. (d) R70013.01 (m) R70013.02 (y)	.cpsflag(5) R70023.
.id(6)	.name(6)	.curflag(6)	.stopdate(6)	.startdate(6)	.cpsflag(6)
.id(7)	.name(7)	.curflag(7)	.stopdate(7)	.startdate(7)	.cpsflag(7)
.id(8)	.name(8)	.curflag(8)	.stopdate(8)	.startdate(8)	.cpsflag(8)
.id(9)	.name(9)	.curflag(9)	.stopdate(9)	.startdate(9)	.cpsflag(9)
.id(10)	.name(10)	.curflag(10)	.stopdate(10)	.startdate(10)	.cpsflag(10)

Appendix 14: Instrument Rosters

BIOCHILD ROSTER (2000)

(**Biological Child CRF**)

PREFIX = "biochild4." (e.g. biochild4.id.01)

ID Number	Sex	Month of Birth	Year of Birth	Status
.id.01 R69638.	.gencode.01 R69708.	.bdate.01~m R69648.01	.bdate.01~y R69648.02	.statcode.01 R69658.
.id.02 R69639.	.gencode.02 R69709.	.bdate.02~m R69649.01	.bdate.02~y R69649.02	.statcode.02 R69659.
.id.03 R69640.	.gencode.03 R69710.	.bdate.03~m R69650.01	.bdate.03~y R69650.02	.statcode.03 R69660.
.id.04 R69641.	.gencode.04 R69711.	.bdate.04~m R69651.01	.bdate.04~y R69651.02	.statcode.04 R69661.
.id.05 R69642.	.gencode.05 R69712.	.bdate.05~m R69652.01	.bdate.05~y R69652.02	.statcode.05 R69662.
.id.06 R69643.	.gencode.06 R69713.	.bdate.06~m R69653.01	.bdate.06~y R69653.02	.statcode.06 R69663.
.id.07 R69644.	.gencode.07 R69714.	.bdate.07~m R69654.01	.bdate.07~y R69654.02	.statcode.07 R69664.
.id.08 R69645.	.gencode.08 R69715.	.bdate.08~m R69655.01	.bdate.08~y R69655.02	.statcode.08 R69665.
.id.09 R69646.	.gencode.09 R69716.	.bdate.09~m R69656.01	.bdate.09~y R69656.02	.statcode.09 R69666.
.id.10 R69647.	.gencode.10 R69717.	.bdate.10~m R69657.01	.bdate.10~y R69657.02	.statcode.10 R69667.

NBIOCHILD ROSTER (2000)

(**Non-biological Child CRF**)

PREFIX = "nbiochild4." (e.g. nbiochild4.id.01)

ID Number	Sex	Month of Birth	Year of Birth	Status
.id.01 R69718.	.gender.01 *	.bdate.01~m R69727.01	.bdate.01~y R69727.02	.statcode.01 R69736.
.id.02 R69719.	.gender.02 *	.bdate.02~m R69728.01	.bdate.02~y R69728.02	.statcode.02 R69737.
.id.03 R69720.	.gender.03 *	.bdate.03~m R69729.01	.bdate.03~y R69729.02	.statcode.03 R69738.
.id.04 R69721.	.gender.04 *	.bdate.04~m R69730.01	.bdate.04~y R69730.02	.statcode.04 R69739.
.id.05 R69722.	.gender.05 *	.bdate.05~m R69731.01	.bdate.05~y R69731.02	.statcode.05 R69740.
.id.06 R69723.	.gender.06 *	.bdate.06~m R69732.01	.bdate.06~y R69732.02	.statcode.06 R69741.
.id.07 R69724.	.gender.07 *	.bdate.07~m R69733.01	.bdate.07~y R69733.02	.statcode.07 R69742.
.id.08 R69725.	.gender.08 *	.bdate.08~m R69734.01	.bdate.08~y R69734.02	.statcode.08 R69743.
.id.09 R69726.	.gender.09 *	.bdate.09~m R69735.01	.bdate.09~y R69735.02	.statcode.09 R69744.

NLSY79 APPENDIX 15:

RECIPIENCY EVENT HISTORIES

RECIPIENCY EVENT HISTORIES

The manner in which information about program recipiency for NLSY79 respondents is collected changed in 1993, the first time all interviews were conducted with an electronic instrument using computer-assisted personal interviewing (CAPI). Information collected in 1993 and after takes the form of a more complete event history, without regard to skipped interviews. The CAPI variables have been combined with the PAPI variables in order to create an event history for each respondent for each of the five types of recipiency (AFDC, Food Stamps, SSI and other public assistance/welfare, unemployment compensation, and spousal unemployment compensation).

These event history variables are located within the RECIPIENCY record type on the main NLSY79 CD. For each type of recipiency, there is (1) a monthly indicator of receipt or non-receipt, (2) a monthly dollar value of the benefits, (3) a yearly indicator of receipt or non-receipt, and (4) a yearly dollar value of the benefits. In addition to these benefit-specific variables, there are also two yearly summary variables which indicate (1) whether or not the respondent has received any benefits from AFDC, Food Stamps, or SSI and other public assistance/welfare, and (2) the dollar value of benefits from all of these sources.

This appendix first contrasts the collection of information on recipiency in the PAPI years to that of the CAPI years. It then describes the creation and editing process for the recipiency event history variables.

PROGRAM RECIPIENCY IN PAPER-AND-PENCIL INTERVIEWS

In paper-and-pencil (PAPI) NLSY79 rounds (1992 and prior), information on R and spouse unemployment compensation, AFDC, Food Stamps and other welfare recipiency was gathered for the calendar year prior to the interview year only. For instance, someone interviewed in 1992 was asked about the months of recipiency in 1991 only. An average figure per week/month was then asked for the entirety of 1991. For example, if a respondent said s/he was receiving AFDC in March, April and May of 1991, and again in September and October of 1991, s/he was only asked for an average amount per month received during those months in 1991.

Data collected in this manner generates a complete event history **only** for respondents who were interviewed at each interview date. For those respondents, information would be present for each month benefits were received from January 1978 through December 1991 (the year before the 1992 interview). However, a respondent skipping one or more interviews would be missing information for each calendar year preceding missed interview years. For example, a respondent missing the 1985 and 1990 interviews would be missing recipiency information for calendar years 1984 and 1989.

PROGRAM RECIPIENCY IN CAPI INTERVIEWS

Beginning with the 1993 CAPI survey, respondents are asked about recipiency in two different manners. First, if the respondent reported receiving benefits in December of the year before the last interview, s/he is then asked if s/he has received continuously since then. If the respondent answers "yes" to this question, s/he is then asked for an average dollar per month/week in each year that benefits were received. The respondent then begins the next section of the interview. Respondents answering "no" are then asked for the date that they first stopped receiving benefits. For these individuals, average dollar values are collected for each year within this initial spell. These respondents are then asked the same questions as the respondents who were not receiving benefits in December of the year preceding the last interview.

Respondents who were not receiving benefits in December of the year before the last interview are asked if they have received benefits at all since January of the last interview year. Individuals who report no recipiency since January of the last interview skip to the next section of the interview. If the respondent answers "yes," s/he is then asked for the date when the benefits first began. This is considered the first

spell. Respondents are then asked if benefits have been received continuously since this start date. If the respondent answers "yes," receipt has been continuous, s/he is asked for the average dollar amount received per month/week. These respondents then proceed to the next section of the interview. If the respondent answers that receipt has not been continuous since this first start date, s/he is asked to report the first date s/he stopped receiving benefits. Average dollar figures per month/week are collected for each year within this first spell.

All respondents who report completing a first spell since January of the last interview are asked if they started receiving benefits again since the first spell ended. Information on up to five spells is collected in the manner described above. If there are more than five spells, the respondent is asked about the first five and the most recent. The flow of questions within these sections is illustrated in Figure A15.1.

In interviews following the initial CAPI interview (1993 in most cases), respondents are asked to verify the last month they reported receiving benefits (if any). They are then asked about any recipiency since their date of last interview. With CAPI, the retrospective recipiency event history is collected from the date of the last interview, providing a more continuous longitudinal record, even for respondents who skip interviews.

VARIABLE CREATION

PAPI Interviews

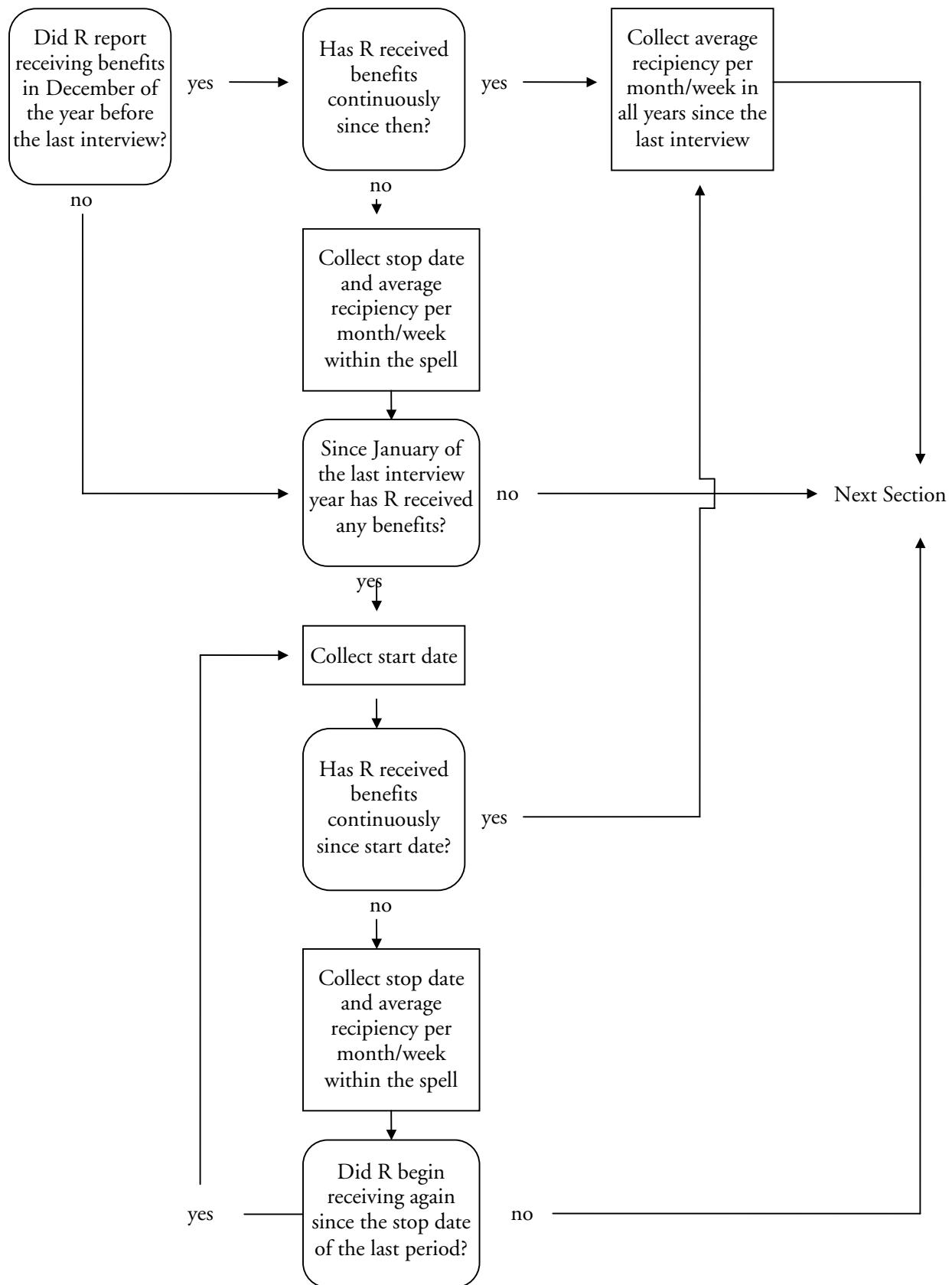
For most of the PAPI years, the yearly and monthly receipt/non-receipt variables are taken directly from responses, and the average monthly value of benefits is used for each month that the respondent reports receiving benefits. For unemployment compensation, weekly averages were collected. This weekly average was multiplied by four and then used as the monthly average. However, there are two main exceptions to this. First, the Food Stamp program underwent a change in 1979. Prior to this, recipients were allowed to purchase food stamps at a price below their market value. Because the 1979 interview asked respondents about recipiency in 1978, respondents who reported receiving food stamps were asked how much they paid for the food stamps in addition to the dollar value of the food stamps received in the last month they received benefits in 1978. The net transfer for 1979 is estimated by subtracting the dollar amount paid from the dollar value received. In all subsequent years, respondents were only asked for the dollar value received in the last month of the previous year that benefits were received.

The second exception concerns SSI and other forms of public assistance/welfare. The series of questions pertaining to public assistance/welfare and SSI has undergone some changes since the beginning of the survey. Initially, in 1979, respondents were asked in a single question if they had received income from any of the sources mentioned above. Respondents were also asked which months benefits were received and the average amount received each month. They were then asked to identify from which sources they received benefits. However, it is not possible to identify how much of this amount is attributable to each source if more than one source was reported.

From 1980 through 1984, the question was divided into two separate ones. Respondents were first asked if they had received any benefits from SSI in the preceding year. They were then asked which months benefits were received and the average amount received each month. A second set of questions asked respondents if they had received public assistance/welfare in the preceding year and, if so, in which months and the average amount received each month.

The format of the questions was changed once again in 1985 and has remained the same since then. As with the initial interview in 1979, respondents were asked if they had received any benefits from SSI, public assistance/welfare. They were then asked which months benefits were received and the average amount received each month. However, unlike the 1979 interview, respondents were not asked to identify the source of the benefits.

Figure A15.1 Flow of Program Recipiency Questions in CAPI Interviews



Because the responses from 1985 through 1992 contain the least amount of information, the other responses need to be converted to resemble responses to the single question asked in those years. Thus, if the respondent reported receiving benefits from either SSI or public assistance/welfare in answer to the separate questions, then the response to the single question becomes yes. The same methodology was used in determining recipiency in each month. If benefits were reported from both questions, then the amount received that month was determined to be the sum of the averages.

CAPI Interviews

Due to the way PAPI interviews collected data (for the calendar year prior to the survey year), information on recipiency is available beginning with January of 1978. Designating this to be month 1 of the monthly event history, all start and stop dates can be identified by their month number. This may be easily calculated using the following algorithm: $\text{month_#} = (\text{year} - 1978) \times 12 + \text{month}$. For instance, June of 1993 would be: $(1993 - 1978) \times 12 + 6 = 186$. Once all start and stop dates have been calculated, the event history for each individual can be created.

To illustrate this, consider Case 1 from Table A15.1. This respondent was not interviewed in 1992 which means that her/his event history from the PAPI years would contain information up through December of 1990. Thus, the beginning month of the CAPI event history would be January of 1991 (month 157). According to the example, this respondent was receiving benefits in December of 1991 and continued to do so until June of 1991 (month 162) and then received no further benefits. The event history would then be formed by placing ones into months 157 - 162 and zeros into months 163 - 186. The dollar amount variable would be created similarly; the dollar value reported for average benefits in 1991, 135, would be placed into months 157 - 162 and zero would be placed into months 163 - 186. This same logic can be applied to each respondent, regardless of the number of reported spells of recipiency: placing ones into all months within a spell (from $\text{start_spell}(i)$ to $\text{stop_spell}(i)$) and zeros into all months outside of spells ($1 + \text{stop_spell}(i)$ to $\text{start_spell}(i+1) - 1$).

To illustrate more completely how each respondent's event history was created, Table A15.1 depicts four additional hypothetical cases. Cases 2 and 4 represent respondents who receive continuously after their start dates; Case 3 depicts a respondent who reports no benefit receipt; and Case 5 represents a respondent who reports two completed spells of recipiency. Table A15.2 presents the event histories which would result if the information had been given by the respondents portrayed in Table A15.1.

In each CAPI interview, information is collected for all time up to the interview date. Because all respondents are not interviewed in the same month, the resultant event histories would be of unequal length. In order to avoid this, a -4 is placed into each monthly indicator and dollar value from the month following the interview month to December of the interview year. These -4's function merely as place savers and will be replaced by information collected in the next interview, or by -5's if the respondent is not interviewed. For example, if the respondent represented by Case 1 is interviewed in September of 1994 and reports no benefit receipt since the last year, then the -4's for July to December of 1993 become 0's and -4's are placed in the months and dollar values for October to December of 1994. These new -4's would later be replaced by information from the 1996 interview.

Table A15.1 Five Hypothetical CAPI Cases

<i>Question</i>	Case				
	1	2	3	4	5
Interview date	6/93	6/93	10/93	7/93	8/93
Year of last interview	1991	1991	1991	1991	1991
Receive Dec year before last interview?	Y	Y	N	N	N
Spell_0 continuous?	N	Y	—	—	—
First stop date spell_0	6/91	—	—	—	—
Average monthly/weekly benefits in '91 months (rec'd Dec or year before last int)	135	—	—	—	—
Receive since Jan of last interview?	—	—	N	Y	Y
Start date spell_1	—	—	—	3/91	3/91
New spell since stop date spell_0	—	—	—	—	—
Start date spell_1	—	—	—	—	—
Spell_1 continuous?	—	—	—	Y	N
Stop date spell_1	—	—	—	—	9/91
Average monthly/weekly benefits in '91 months (1 st new spell)	—	—	—	—	200
New spell since stop date spell_1?	—	—	—	—	Y
Start date spell_2	—	—	—	—	2/93
Spell_2 continuous?	—	—	—	—	N
Stop date spell_2	—	—	—	—	5/93
Average monthly/weekly benefits in '93 months (2 nd new spell)	—	—	—	—	225
New spell since stop date spell_2?	—	—	—	—	N
Average monthly/weekly benefits in '91 months (rec'd contr'nly since last start date)	—	145	—	157	—
Average monthly/weekly benefits in '92 months (rec'd contr'nly since last start date)	—	152	—	160	—
Average monthly/weekly benefits in '93 months (rec'd contr'nly since last start date)	—	175	—	163	—

Table A15.2 Resultant Event Histories

	Case									
	1		2		3		4		5	
	yes/no	dollar								
1/91	1	135	1	145	0	0	0	0	0	0
2/91	1	135	1	145	0	0	0	0	0	0
3/91	1	135	1	145	0	0	1	157	1	200
4/91	1	135	1	145	0	0	1	157	1	200
5/91	1	135	1	145	0	0	1	157	1	200
6/91	1	135	1	145	0	0	1	157	1	200
7/91	0	0	1	145	0	0	1	157	1	200
8/91	0	0	1	145	0	0	1	157	1	200
9/91	0	0	1	145	0	0	1	157	1	200
10/91	0	0	1	145	0	0	1	157	0	0
11/91	0	0	1	145	0	0	1	157	0	0
12/91	0	0	1	145	0	0	1	157	0	0
1/92	0	0	1	152	0	0	1	160	0	0
2/92	0	0	1	152	0	0	1	160	0	0
3/92	0	0	1	152	0	0	1	160	0	0
4/92	0	0	1	152	0	0	1	160	0	0
5/92	0	0	1	152	0	0	1	160	0	0
6/92	0	0	1	152	0	0	1	160	0	0
7/92	0	0	1	152	0	0	1	160	0	0
8/92	0	0	1	152	0	0	1	160	0	0
9/92	0	0	1	152	0	0	1	160	0	0
10/92	0	0	1	152	0	0	1	160	0	0
11/92	0	0	1	152	0	0	1	160	0	0
12/92	0	0	1	152	0	0	1	160	0	0
1/93	0	0	1	175	0	0	1	163	0	0
2/93	0	0	1	175	0	0	1	163	1	225
3/93	0	0	1	175	0	0	1	163	1	225
4/93	0	0	1	175	0	0	1	163	1	225
5/93	0	0	1	175	0	0	1	163	1	225
6/93	0	0	1	175	0	0	1	163	0	0
7/93	-4	-4	-4	-4	0	0	1	163	0	0
8/93	-4	-4	-4	-4	0	0	-4	-4	0	0
9/93	-4	-4	-4	-4	0	0	-4	-4	-4	-4
10/93	-4	-4	-4	-4	0	0	-4	-4	-4	-4
11/93	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4
12/93	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

Handling Don't Knows and Refusals

In PAPI years when the respondent did not know whether s/he had received benefits in the previous year, a “-2” was placed in all months and dollar values for that year. For respondents who refused to answer this question, “-1” was entered into all months and dollar values for that year.

In CAPI years, when asked for the start or stop date of a spell, a respondent could respond “don’t know.” When the respondent does not know (or refuses to answer) the start date of a spell of recipiency, s/he is then asked approximately how many months/weeks s/he received benefits and how much s/he received in the last month/week s/he received benefits. If a respondent does not know the start date and there are valid responses for these questions (i.e., responses greater than zero), the start date is set at the first possible point of the unfilled event history and “-2” is placed into the number of months that the respondent reports receiving. For example, if a respondent last interviewed in 1990 and being interviewed in 1993 responds that s/he has received benefits since January of the last interview year but does not know when s/he started receiving, the start date is set at January of 1990. If this same respondent reports that s/he received benefits for six months and received \$200 the last month s/he received benefits, then “-2” would be filled into the January through June of 1990 receipt/non-receipt variables and \$200 would be filled into the dollar values for these months. The receipt/non-receipt and dollar value monthly variable for July 1990 through the interview date would then be filled with zeros. If the respondent does not know the stop date but has reported a start date, the same logic is employed using the reported start date.

EDIT FLAGS

The dollar values elicited by the questions in the recipiency section of the questionnaires were meant to be monthly (weekly in the case of unemployment compensation) averages. However, some of the responses appear to be unrealistically high or low. When responses seem unreasonably high, one of the most likely reasons is that the respondent has reported a total value as opposed to a monthly/weekly average. When the event histories were created, all reported dollar values were first tested for their plausibility. Values which were determined to be “wrong” were then edited. The following describes the process by which the reported values were tested and assigned a *yearly* edit flag. Although the meaning of the edit flags is the same in PAPI years as in CAPI years, the editing process was slightly different for the two. Edit flag meanings are summarized in Table A15.3.

PAPI Years

For each year, average values were collected for each program’s recipiency rates. From the average values, an “acceptable” range was constructed. In general, this range is constructed by setting the lower bound to be a fraction of the average and the upper bound to be a multiple of the average. Table A15.4 reports the ranges which were used in each year for each program. All dollar values which fell within the ranges defined in the table below were assumed to be correctly reported. These observations were left unedited and assigned an edit flag of “0.” Reported values which fell below the lower bound were denoted with an edit flag of “1,” but they were left unedited as there is no obvious explanation for these low figures.

There were some observations where the individual reported an average which was greater than 10,000 per month/week. These observations have been top-coded at 9996. In editing, these observations were not transformed; both monthly and yearly dollar values were left at 9996. These observations are identifiable by an edit flag of “6.”

Table A15.3 Edit Flag Values for Recipiency Data

Value	Meaning of Flag
0	reported value within “acceptable” ranges— <i>reported value</i> used
1	reported value too low— <i>reported value</i> used
2	reported value too high but calculated value not within acceptable ranges— <i>reported value</i> used
3	reported value too high— <i>calculated value</i> used
4	reported value high but within range of preceding or following year— <i>reported value</i> used
5	reported value too high but calculated values not within range, however calculated value is within acceptable ranges of preceding or following year— <i>calculated value</i> used
6	9996 top-code
7	flag for SSI variable not equal to PA variable (only used in 1980-1984 interview data)
9	(applies only to food stamps 1979 interview data variable) amount paid for stamps is known but the value is unknown or values is known but amount paid for stamps is unknown—dollar value set to -2
10	(applies only to food stamps 1979 interview data variable) reported amount paid for the stamps is greater than the reported value of the stamps—dollar value set to -2

Because data on Food Stamps in 1979 and SSI between 1980 and 1984 were collected differently (see the discussion in the section on Variable Creation in PAPI Interviews), there are some values of the edit flags which pertain specifically to them. If the editing which was required for the SSI variable was different from the editing required for the “public assistance and other welfare” variable, then the edit flag was set to “7.” If the respondent reported the value of the Food Stamps but not the cost or reported the cost but not the value, the net transfer could not be determined. In these cases, the dollar values were set equal to “-2” and have edit flags equal to “9.” There were also a small number of cases where the respondent reported a cost which was greater than the value of the Food Stamps. Again, the dollar values were set equal to “-2,” but the edit flags were set equal to “10.”

In the PAPI years, the total number of months that benefits were received were summed together. If the reported average monthly value was above the upper range, then a new average value was calculated by dividing the reported value by the total number of months that benefits were received. If this new figure fell above the lower bound, it was then assigned to be the monthly benefit and these observations were assigned an edit flag of “3.” If the calculated average was less than the lower bound, the original reported figure was assigned to be the monthly benefit and these observations were assigned an edit flag of “2.”

Two additional checks were performed on the PAPI data to ensure that the transformed data seemed consistent with values reported from other years. Observations whose calculated averages were too low (edit flags of “2”) were compared to the value reported in the year immediately preceding and following them. If the computed value fell within 50% to 150% of the values reported in either the year preceding or following it, then the observation’s monthly benefit was changed to the computed value, and the edit flag was set equal to “5.” Observations whose calculated averages were above the lower bound (edit flag of “3”) were compared to the value reported in the year immediately preceding and following them. If the reported value fell within 50% to 150% of the values reported in either the year preceding or following it, then the reported value was assigned to be the monthly value, and the edit flag was set equal to “4.”

For example, suppose a respondent reported that s/he received AFDC in January, February, and March of 1985 and the average monthly benefit was \$750. This value is above the upper bound of \$726 for 1985. The algorithm directs that the reported monthly benefit, \$750, be divided by the total number of months benefits are received, 3. The resultant figure, \$250, does fall within the acceptable ranges and would be assigned to each month, yielding a yearly total of \$750. These observations would be assigned an edit flag of "3." However, this reported value would then also be compared to AFDC recipiency reported in 1984 and 1986. Suppose that this same respondent reported receiving AFDC in January through December of 1984 with an average monthly benefit of \$725. Because this value is above the upper bound of \$700 for 1984, an average value would be calculated for this year as well. However, the resultant average of \$60 is less than the lower bound for this year. Each month would have been assigned a value of \$725, and an edit flag of "2" would have been assigned for the year. When the reported value from 1984 is compared to that reported in 1985, it seems likely that the \$750 reported in 1985 was the correct average. Thus, the monthly event history for this respondent would assign a value of \$725 for all months in 1984 and \$750 for January, February, and March of 1985. The edit flag would be "2" in 1984 and "4" in 1985.

Table A15.4 Acceptable Ranges: 1978 - 2000

	AFDC		Food Stamps		SSI		Unemployment Compensation	
	lower	upper	lower	upper	lower	upper	lower	upper
1978	10	537	21	244	79	631	36	158
1979	10	579	24	275	85	682	40	173
1980	10	586	28	316	93	742	43	187
1981	10	622	27	344	100	796	48	208
1982	10	643	30	342	107	858	50	217
1983	10	670	30	360	111	888	49	215
1984	10	700	31	364	114	915	51	222
1985	10	726	32	366	123	988	54	236
1986	10	742	35	400	127	1017	56	245
1987	10	771	36	415	132	1052	58	254
1988	10	794	41	478	139	1111	61	266
1989	10	815	45	511	152	1213	65	284
1990	10	809	48	549	162	1298	68	298
1991	10	786	48	544	181	1447	70	305
1992	10	875	48	550	193	1547	70	305
1993	10	844	48	555	190	1519	72	315
1994	10	827	49	560	195	1561	74	322
1995	10	808	50	568	200	1603	76	333
1996	10	808	50	568	200	1603	76	333
1998	10	949	63	735	210	1700	88	383
2000	10	949	63	735	210	1700	88	383

CAPI Years

For the CAPI years the reported value was filled into each month the respondent reported receiving benefits. The individual months were summed up and the totals were tested. The acceptable ranges were calculated by multiplying the upper and lower bounds by the number of months within the year that the

respondent reported receiving benefits. If the value fell within the acceptable range then the year was assigned a flag of "0." If the total fell below the lower bound, then the year was assigned a flag of "1." If the total fell above the upper bound then an average was calculated by dividing the total by the number of months received. If the average was within the acceptable range, it was then assigned to be the yearly total. In addition, each month's value was divided by the total number of months. These observations are assigned edit flags of "3."

For example, consider a person whose monthly event history for AFDC for 1995 looks like the one below:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
0	0	2500	2500	2500	2500	2500	0	2500	2500	2500	2500	22500

This respondent reported nine months of receipt so the range of acceptable values would be \$846 (9×94) to \$7,272 (9×808). Because the \$22,500 yearly total is above the calculated upper bound, an average is taken. The resultant \$2,500 ($22,500/9$) does fall within the acceptable range and would become the yearly value. For 1995 for AFDC, this respondent would have an edit flag of "3" and the following event history:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
0	0	278	278	278	278	278	0	278	278	278	278	2500

The additional comparison across years that was made on the PAPI data was not made on the CAPI data. In the CAPI years, individuals are often reporting multiple years within the same interview. If a respondent misinterprets the question the first time when asked to report the average amount received per month/week for one year, s/he is likely to do so for each year that s/he reports within a survey. Thus, it is likely that multiple values reported within a single interview will be similar from year to year and comparing them would not be useful.

FILL FLAGS

Because the new format of the CAPI questionnaire allows information to be collected from years when an individual had missed an interview, the number of noninterviews (-5's) appearing in the recipiency event history will not match the number of "official" noninterviews for the year. This is due to the fact that CAPI interviews collect information dating back to the last interview date. Thus, when a respondent is interviewed after missing an interview in the CAPI years, the -5's are replaced by the actual information. This same issue is encountered in the Workhistory data. In order to identify the "true" noninterviews in each year, a fill flag has been created. This is a monthly variable which indicates the interview year from which the information was collected. All data collected from PAPI interviews is identified with a fill flag of "0." For any of the PAPI years, the data will have come from the interview after that year, i.e., if March 1985 has a fill value of "0," then the data came from the 1986 interview. In the CAPI years, each year has been assigned a different value. Table A15.5 summarizes the values of these fill flags.

Table A15.5 Fill Flag Values: 1978 - 1996

Flag Value	Interview Year
0	PAPI data (1979 - 1992)
1	1993
2	1994
3	1996

Respondents with fill values higher than the flag value for the interview year have had data backfilled into them. For example, any observation from 1978 to 1991 that has a fill flag greater than zero has been backfilled.

FILL FLAGS FOR 1998

These flags contain data for all respondents, regardless of whether they were interviewed in a given survey year. They indicate either the survey period/year in which data for that respondent was collected, or the last survey period/year in which that respondent was interviewed. These flags correct the instances where the data appears as a -4 (valid skip- question not asked) but should actually be 0, indicating that the question was asked but no recipiency was reported. The flags and their definitions are listed below.

NONINT-FLAGS (flag to id non-interviews in created recipiency variables for a given year)

A code of '1' on this flag indicates that created recipiency variables for this year should be recoded to '-5'. This means that no interview has been conducted that would have collected information for this calendar year.

VMSNG-FLAGS (flag to id cases that were valid missings before '0's were recoded to '-4's aswell for all recipiency but spouse unemployment variables)

This flag applies to all amount variables except those pertaining to spouse unemployment (see SVMSNG-FLAGS) '0' values in the monthly and yearly amount of receipt variables were recoded to -4 for public release. These '0' values indicated respondents who were asked the questions and reported no receipt. Other respondents who were not asked the questions were coded '-4' and retained that code for public release. A code of '1' on this flag indicates that the respondent was always coded '-4' and should remain so. All other monthly and yearly amount of receipt variables can be recoded to '0' if users wish.

SVMSNG-FLAGS (flag to id cases that were valid missings before '0's were recoded to '-4's aswell for spouse unemployment variables)

This flag applies only to amount variables pertaining to spouse unemployment. (See VMSNG-FLAGS for flag pertaining to other amount variables). '0' values in the monthly and yearly amount of receipt variables were recoded to -4 for public release. These '0' values indicated respondents who were asked the questions and reported no receipt. Other respondents who were not asked the questions were coded '-4' and retained that code for public release. A code of '1' on this flag indicates that the respondent was always coded '-4' and should remain so. All other monthly and yearly amount of receipt variables can be recoded to '0' if users wish. This flag applies to all amount variables except those pertaining to spouse unemployment (see SVMSNG-FLAGS).

These flags were created for the 1998 release only.

NLSY79 APPENDIX 16:
1994 RECALL EXPERIMENT

THE RECALL EXPERIMENT

Beginning with the 1996 survey, the NLSY79 became a biennial survey. In anticipation of changing to a 2-year interview period, an experiment dubbed the “Recall Experiment” was conducted in 1994 on a portion of the eligible sample. A subsample was drawn from the members of the original 12,686 sample still eligible for interview in 1994, who were also interviewed in 1992 and 1993. This subsample was treated as if their 1993 interview never took place; their date of last interview was established as the 1992 interview date. The information that drove the 1994 interview was that gathered in the 1992 interview. The affected respondents were periodically reminded where applicable, that the reference date for their interview was not the 1993 interview date, but the 1992 interview date.

The result for the 854 “recall respondents” interviewed in 1994 is that retrospective information, pertaining mainly to the period since the last interview, was essentially re-reported for the period between the 1992 and 1993 interviews, in addition to the new information for the 1993–94 survey period. The re-reported information for the 1992–93 interview period can be compared to that previously reported during the 1993 interview for possible discrepancies.

The goal of this experiment was to gain a better sense of the possible consequences for respondent recall, of a biennial instead of annual survey administration, for accuracy and consistency of data and overall respondent burden of participating in the survey. Certain segments of NLSY79 surveys in specific years have previously collected retrospectives over a 2-year period or longer. For example the more detailed two-year fertility history sponsored by The National Institute for Child Health and Development (NICHD) has been administered in selected survey years. However, except in the case of a respondent who actually skips one or more interviews, retrospectives in most segments of the questionnaire for a given year, have required that a respondent only recall events and circumstances over the period of roughly a year.

EFFECTS ON NLSY79 DATA

Because the information for the 1992–93 interview period was reported twice, during both the 1993 and 1994 interviews, users may encounter some degree of difference and inconsistency in that data gathered in 1993 and 1994. Data pertaining to past calendar year, such as spouse’s labor force activity or respondent’s income and assets, is not affected by the experiment. Retrospectives for which data between the 1992 and 1993 interview period would have been reported both in the 1993 and 1994 interviews is listed below.

Marital History

Retrospective/event history of changes in marital status since date of last interview

Regular Schooling

Two year retrospective of specific months of enrollment (if any) in regular school for the 1993 calendar year only

Retrospective/event history of college attendance since date of last interview

Military

Retrospective/event history of military enlistment and separation dates since date of last interview

On Jobs/Employer Supplements

Retrospective/event history of employment with specific employers, and periods not working for specific employers (gaps within jobs), since date of last interview

Gaps

Retrospective/event history of periods of non-employment (gaps between employers), since date of last interview

Training

Retrospective/event history of (continued) participation since the last interview, in training programs either reported at the date of last interview or enrolled in since the date of last interview

Health

Retrospective/event history of most recent and most severe work-related injuries (if any), since date of last interview

Income and Assets

Retrospective/event history of program recipiency (respondent/spouse unemployment compensation, AFDC, government food stamps, SSI/other welfare) since either December, 1992 if receipt reported during that month, or January 1993 for those not reporting receipt in December 1992

The fertility history section is a special case with respect to the Recall Experiment. Between the 1986 and 1992 interview years, in which paper-and-pencil interviewing (PAPI) was used, information on biological children was collected in each odd-numbered year on any biological children born since the date of last interview. However, even-numbered years contained an expanded fertility history section, sponsored by NICHD. This expanded history included a re-reporting of biological children born since the date of the last NICHD interview (even-numbered year). Newly reported children were handwritten onto the records when paper-and-pencil instruments were being used. This allowed interviewers to easily identify children about whom certain series of questions should be asked. The fertility data therefore, has for years contained a limited version of the 1994 Recall Experiment, specific to biological children. Other segments of the fertility history (pregnancy information for female respondents and visitation habits of biological children with non-residential parents) were collected exclusively as two-year retrospectives in even-numbered years. With the advent of CAPI interviewing, it was decided that re-reporting of biological children in NICHD years was no longer necessary. Interviewers no longer needed to rely on visual identification of children reported since the last NICHD interview, because these children could be mechanically flagged. Therefore, in 1994, respondents not belonging to the Recall Experiment sample were for the first time since 1986, only required to report new children born since the date of last interview, instead of the last NICHD interview (two years ago in most cases). However, Recall Experiment respondents were asked to update their biological child records since the last NICHD interview year (1992), as was the norm in even-numbered years since 1986. This will continue to be the case for all respondents with the biennial administration, which began in 1996. However, the elimination of the odd-year survey also means the continued elimination of re-reporting of children born since the last NICHD interview.

EFFECTS ON AUXILIARY DATA FILES AND VARIABLES IN NLSY79

The potential seam effects introduced by the Recall Experiment are particularly relevant for the Work History data file and the creation of the key variable, Total Net Family Income. In each case, a different procedure was used to eliminate possible discrepancies in information reported both during the 1993 and 1994 interviews.

Work History 1979-1994 Data File

The 1979–93 release of the work history data file already incorporated information covering the period between the 1992–93 interviews in each respondent's longitudinal labor force history. The possible disruptions that might occur to the longitudinal record if inconsistent data were introduced into the formulas for a single time frame were unpredictable and potentially serious. It was determined that information covering the period between the 1992 and 1993 interview, reported by the recall respondents in the 1994 interview, would be eliminated for the purposes of creating the 1979–94 work history file. The following basic decision rules were applied:

- 1) Data pertaining to employers, gaps in employment and/or periods of military service for which the start and stop dates fell completely prior to the 1993 interview date for recall respondents were completely eliminated for the purposes of creating the 1979–94 work history data file. These should have been reported during the original 1993 interview;
- 2) Data pertaining to employers, gaps in employment and/or periods of military service for which the start and stop dates fell completely within the period between the 1993 and 1994 interview were retained in their entirety. These are constitute new information and would not have been reported during the 1993 interview;
- 3) Data pertaining to employers, gaps in employment and/or periods of military service for which the start and stop dates fell partially prior and partially after the 1993 interview date, was truncated where necessary at the 1993 interview date (e.g. start dates of these periods of employment/non-employment were set to the 1993 interview date).

Users should note that, with respect to gaps in employment, the NLSY79 does not establish specific weeks when a respondent might be looking for work or laid off (making him/her unemployed as opposed to out of the labor force or OLF). Respondents report only a total number of weeks during each gap for each status. This made it impossible to determine which segment of the 1992–94 interview period should be assigned the unemployed code and which the OLF code for gaps which fell partially prior and partially after the 1993 interview, and in which the respondent reported both statuses.

Total Net Family Income

For the purposes of creating Total Net Family Income, the total number of weeks/months of program recipiency in the past calendar year (1993), and the average amount received per week/month in 1993 are required. Recall respondents would have reported all 1993 recipiency in the 1994 interview, not just that occurring prior to the 1993 interview date. Therefore, 1994 data were used exclusively to compute the total 1993 recipiency figures for the recall respondents. This avoided the task of attempting to combine data from 1993 and 1994 for these respondents and deal with potential inconsistencies in reports between the two interviews.

NLSY79 APPENDIX 17:

ISIS DATA

ISIS Data

Many researchers are interested in knowing if or how much interviewers affect respondent's answers. To enable researchers to investigate these questions the NLSY79 data release contains information on each interviewer's characteristics. Information on the characteristics of NLSY79 interviewers comes from the National Opinion Research Center's (NORC) interviewer personnel files, which are called ISIS.

For each survey (1979–2000) the following ISIS variables are available: ISIS interviewer ID, number of times this interviewer has already interviewed the respondent, the interviewer's race, the interviewer's sex, the interviewer's age, the interviewer's educational level, the amount of time the interviewer has worked at NORC, the languages other than English spoken by the interviewer, a flag identifying if the interviewer was ever terminated and the last known pay rate (in U.S. dollars per hour) for the interviewer. For the surveys run from 1981 to 1985 two additional variables are available; the number of NORC surveys the interviewer worked on during the calendar year and the number of hours spent interviewing during the calendar year.

Researchers should note that CHRR was provided with only a single snap shot of the ISIS data base. This snapshot reflects the ISIS data base status a few months prior to the fielding of the NLS 1994 survey. Hence, changes in items like an interviewer's educational attainment or pay rate are not reflected in the data.

Constructing the ISIS ID

The key variable which links the NLSY79 data set with the ISIS data set is the ISIS Interviewer ID. This variable is similar but not exactly the same as the Interviewer ID variable which has been found on the NLSY79 public use data set for many years. The ISIS Interviewer ID is constructed from the NLSY79 Interviewer ID variable using the following steps.

First, the NLSY79 interviewer ID for all years prior to 1996 is divided by ten to truncate the last digit. This last digit was used to cluster interviewers together and the digit was not used in the ISIS data base. IDs for 1996 do not have this last digit.

Second, each ISIS ID was then run it through the list of all known interviewers who changed their ID. Interviewers changed their ID if they moved to a different states, were promoted or demoted. Only a partial list of interviewers who changed their ID is available so there is no year when the characteristics of all interviewers are known. However, even if all the characteristics of an interviewer are not known, every effort was made to create a consistent ID number since a researcher might be interested in knowing who interviewed whom from year to year even if other information like sex is not available..

The resulting ID was then used to search the ISIS data base. Most of the IDs match but a small number of Interviewer Ids do not. Table 1 below shows the number and percentage of respondents whose interviewer was found in the ISIS data base by survey year. Readers should note that most interviewers interviewed multiple respondents so not finding even a single interviewer in the ISIS data base causes dramatic changes in the number and percentage of interviewers found columns.

Two years (1979 and 1987) have large numbers of interviewers who were not found in the ISIS data base. 1979 has a large number because the ISIS data base was not started until 1980. Any interviewer who worked in 1979 but not in a later year was never added to ISIS. The 1987 interview was primarily fielded via the telephone to save money. Telephone interviewers who did not also do face-to-face interviews are not part of the ISIS data base.

Table 1. Interviewers Identified in ISIS by Survey Year

Year	# of Respondents Interviewed	# of Interviewers Matched in ISIS	Percentage Not Matched in ISIS
1979	12686	9619	24.2%
1980	12141	10946	9.8%
1981	12195	10967	10.1%
1982	12123	11400	6.0%
1983	12221	11980	2.0%
1984	12069	11585	4.0%
1985	10894	10374	4.8%
1986	10655	10119	5.0%
1987	10485	6825	34.9%
1988	10465	9394	10.2%
1989	10605	9906	6.6%
1990	10436	9321	10.7%
1991	9018	8105	10.1%
1992	9016	8480	5.9%
1993	9011	8618	4.4%
1994	8891	8701	2.1%
1996	8636	8050	6.8%
1998	8399	7736	7.9%
2000	8033	7814	2.7%
Average			8.9%

Other ISIS Variables:

This section describes all the other ISIS variables available beyond the Interviewer's ID.

Interviewer Count

This counts the number of years the interviewer has interviewed the respondent. For example if the interviewer count in 1986 contains a 4 this means that the 1986 interviewer for the respondent has already interviewed the person 4 previous times. Please note that the number of times does not include the 1986 interview. Also note the series does not look ahead into the future. If they are also interviewed by the same interviewer in 1987, this information is NOT used to calculate the 1986 number. All 1979 values are zero, since no one has been previously interviewed.

Interviewer Race

- 1 = White
- 2 = Black
- 3 = Hispanic
- 4 = Asian
- 5 = American Indian
- 3 = missing

Interviewer Sex

- 1 = Male
- 2 = Female
- 3 = missing

Interviewer Age

Age of the interviewer in the interview year.
-3 = missing

Interviewer Education

- 1 = Grade 0-8
- 2 = Grade 9-11
- 3 = High School Graduate
- 4 = Vocational degree
- 5 = Some College
- 6 = College Graduate
- 7 = Graduate School
- 8 = Masters Degree
- 9 = Professional Degree
- 0 = Other

Interviewer Tenure

Number of years spent working at NORC as of interview year.

Interviewer Languages

- 1 = Spanish
- 2 = German
- 3 = Italian
- 4 = Polish
- 5 = Japanese
- 6 = Chinese
- 7 = Yiddish
- 8 = American Indian
- 9 = Portuguese

Interviewer Type of Termination

- 0 - Never Left NORC
- 1 - Terminated by NORC for cause
- 2 - Left NORC by own choice

Interviewer Last Pay Rate (at time of ISIS dump)

- 1 \$3.00 to \$5.00 per hour
- 2 \$5.01 to \$7.50
- 3 \$7.51 to \$10.00
- 4 \$10.01 to \$15.00
- 5 \$15.01 and up
- 3 Pay Rate Not in ISIS

For 1981 to 1985 the following two variables also exist:

Hours

Number of hours worked on NORC projects in calendar year

Count

Number of NORC surveys/projects worked on in calendar year

NLSY79 APPENDIX 18:
WORK HISTORY DATA

DESCRIPTION OF THE 1979-2000 NLSY79 WORK HISTORY PROGRAM

This document provides a general description of the work history program and explains the procedures and the logic that went into the creation of the various arrays in the program. The variables in the work history file are the output of the work history program. While the discussion that follows assumes some knowledge of the program and the acronyms that are used in it, this description provides useful information for all users of the work history data.

The work history program was originally written to create the key work variables like "Number of Weeks Worked since Date of Last Interview," "Number of Weeks Worked in Last Calendar Year," etc. These key variables use all recorded jobs for each respondent (up to 10 jobs). The WEEKLY LABOR STATUS, HOURS WORKED, and DUAL JOBS arrays also were created with data from up to 10 jobs for each respondent. However, only 1% of all respondents have more than 5 jobs in any given survey year, resulting in valid missing data for jobs 6 through 10 for 99% of the sample. In order to reduce the total number of variables, the data file contains the JOB variables for only 5 jobs for each respondent.

The purpose of the WEEKLY LABOR STATUS, HOURS WORKED and DUAL JOBS arrays is to create a longitudinal work history record for each respondent through the 2000 interview date. Because each year's survey collects information on jobs held and periods not working since the date of the last interview, it is possible to construct a continuous, week-by-week record for each respondent.

There are a few exceptions, however. In the 1979 and 1980 surveys, job information was collected only for respondents age 16 and older at the date of the interview. Additionally, the 1979 survey data contain the most cases with inconsistent or invalid employment-related data of any survey year, resulting in a greater proportion of missing gaps in the work history record. For example, in 1979 there are 86 cases that have job dates that exceed the interview date; in 1980, there are 11 cases that have job dates that exceed the interview date; in 1981 there are none.

Users should also note that 1,079 members of the military sample were dropped as of the 1985 survey. In 1991, all members of the economically disadvantaged non-black/non-Hispanic oversample were dropped as well. More information on these sample types is available in chapter 2 of the *NLSY79 User's Guide*.

Description of the 1979–94 PL/I Program

The following is an abbreviated step-by-step description of the 1979-1994 PL/I program. In 1996, the PL/I program was converted to SQL code that replicates the PL/I program and functions. See the section below titled "Changes between the 1979–94 and the 1979–96 Work History Data" for more information.

1. All of the variables used in the program are declared and most are included in the PL/I structure called VARIABLES.
2. The variables common to all respondents, like ID, SAMPLE_ID, etc. are assigned values. The week-by-week arrays are initialized to zero and all of the variables included in the WORK_HISTORY part of the structure are initialized to -4.
3. For each interview year, procedures (VARIABLES1979, VARIABLES1980, etc.) that assign the variables for each survey year are called if the respondent was interviewed. Start and stop dates for jobs and periods not working are sent to the WEEK procedure, where the valid month, day and year variables are converted to a week number, with week 1 being January 1, 1978. If the respondent was not interviewed, then all WORK_HISTORY variables for that survey year are set to -5.

4. After all VARIABLES19XX are assigned, the procedure CALC is called to evaluate the various start and stop dates, to assign codes, and to create the job number for all of the jobs for each interview year. Within CALC, the procedure FILL is called to fill in the codes that are assigned to the WEEKLY LABOR STATUS and DUAL JOBS arrays and to calculate the hours worked during each week that are loaded into the HOURS WORKED array.
5. Finally, the procedure SUMMER is called to calculate and sum the key work history variables.

CALC Procedure

This procedure processes all jobs for each survey year, beginning with the first job. CALC starts by calculating each year the number of jobs since the date of the last interview, assigning a job number, and calculating the hourly wage for each job. If the respondent had the job at the date of the last interview, the start date becomes the date of the last interview, which is then “ceiled” or rounded up using the “ceil” function. Next, if the respondent is currently working at the job, it assigns the interview date, which is “floored” or rounded down using the “floor” function, as the stop date. (All dates at this point have been converted to week numbers in the WEEK procedure.)

If the start and stop dates of the job are valid and do not coincide with an interview date, the start and stop dates are “ceiled”. The number of weeks tenure on the job is calculated by subtracting the start date from the stop date of the job. FILL is then called to fill in the week arrays for the particular job. The start and stop dates of the job, the job number, and the number of hours usually worked per week (HOURSWEEK) at the job are sent to the FILL procedure.

If the job had any periods not working associated with it, then each of the four possible periods not working for the employer is processed. If the start and stop dates for the periods not working are valid, a code is assigned indicating whether the respondent was out of the labor force (OLF) or unemployed for the period. If the respondent is OLF the whole period, a code of 4 is assigned. If the period not working is divided between OLF and unemployed, a temporary code of 9 is assigned and the number of weeks unemployed is determined. If the start and stop dates of the period are valid, but the labor force status cannot be determined, a code of 2 is assigned.

The period start and stop dates, CODE, and HOURSWEEK are sent to FILL. If the period dates are invalid, a code of 3 is assigned and start and stop dates of the job are passed to FILL, along with HOURSWEEK. This is only done for the first period not working for the first employer this week.

Next, tenure at the job is again calculated, this time in terms of total weeks on the job instead of just since the date of the last interview. First, a determination is made to see if the employer is the same employer a respondent reported at the time of the previous interview. If there is a previous employer number and the tenure for that previous employer is valid, then the tenure for the job from the previous interview is added to the tenure for the job being processed. Only tenure with an employer that is reported during contiguous survey years can be calculated over the total time spent with an employer. For example, consider a respondent who was interviewed in 1981, 1982 and 1983 surveys. Now suppose the respondent reported having worked for the Labor Department at the time of the 1981 survey and left and then began working for that same employer again by the time of the 1983 survey. Because the employer numbers are only followed between contiguous interviews, there is no way to calculate total tenure with the Labor Department since the respondent did not report that employer during the 1982 survey. Only employers from the previous year's survey are compared with employers reported in the current year's survey.

Finally, CALC evaluates the possible six periods not working or in the military between jobs. For each of the periods not working, the same logic used for the periods not working on a job is used for the periods between jobs.

User Notes: A few additional notes are in order:

1. If the start and stop dates for a job are invalid, then that job has no dates that can be sent to FILL. As a result, there is no record of that job in the WEEKLY LABOR STATUS array and no indication that the job is missing. In 1979, there were 1190 cases with any invalid start or stop dates (i.e., at least one week is unaccounted for – WEEKLY LABOR STATUS=0); in 1980, there were 942 cases; in 1981, there were 254; and in each of the following survey years, there were fewer than 200 cases.
2. A job held in any day of a week is counted as a job for the whole week. This is achieved by “flooring” start dates and “ceiling” stop job dates to integer week values. There is one exception previously mentioned—stop dates for jobs held at the interview date are floored. This is done to avoid double counting across interview years.
3. Start and stop dates for periods not working either with the job or between jobs are “ceiled” in FILL.
4. The HOURS WORKED array is set to -3 if any job in the week has an invalid value for HOURSWEEK. Between 1979 and 1992, the maximum number of hours for any given week is 96. Beginning in 1993, the maximum number of hours for a given week can be reported up to 168 hours (the total number of hours possible in a single week).

FILL Procedure

The FILL procedure takes the start and stop dates that have been converted to week number values and fills in values for the WEEKLY LABOR STATUS, HOURS WORKED and DUAL JOBS arrays for each week between stopping and starting dates that are passed to it.

In FILL, the STATUS array is loaded with either a survey year job number or a code signifying that there was not a civilian job that week (a code of 0, 2, 3, 4, 5, or 7). The DUAL JOBS array is loaded with a survey year job number(s) if more than a civilian job is held that week; otherwise it has a value of zero. The HOURS WORKED array is loaded with the number of hours worked on all jobs held that week, up to a maximum of 96.

FILL is called from the CALC procedure for all start and stop dates except for military start and stop dates. Military start and stop dates are determined in the VARIABLES procedures for each year, and FILL is called from those procedures to fill in a code of 7 in the WEEKLY LABOR STATUS array for active military service.

Initially, FILL checks for valid start and stop dates. If the dates are valid, then FILL takes one of three paths. The first path is to evaluate the WEEKLY LABOR STATUS array for that week to see (1) if it contains a job number, (2) if the code passed from CALC is a job number, and (3) if the previous employer number for the job is different from the job number in the WEEKLY LABOR STATUS array. If all of these statements are true, then FILL determines that the job is not a duplication of the job that exists in the WEEKLY LABOR STATUS array for that week.

Next, FILL looks at the DUAL JOBS array to see if there is a job number in DUAL JOBS. If DUAL JOBS already has a job number(s), then the current job number is compared to the job number(s) in DUAL JOBS. If the job number does not exist in DUAL JOBS, then the HOURSWEEK for that job is added to the number of hours for that week and the job number is added to DUA LJOBS. If the job is a duplicate job, then nothing is done to the arrays.

The second path is taken if there is no dual job and if the week dates are associated with a job or if there is not job number in the WEEKLY LABOR STATUS array. If this is the case, FILL tests for two

conditions. The first condition is met if COD is 9. (A code of 9 means that the respondent had a period not working that was part OLF and part unemployed.) If COD equals 9, then the HOURSWEEK are subtracted from the hours in HOURS, because the respondent is not working at the job. The number of weeks unemployed (code of 4) is arbitrarily assigned to the middle portion of the weeks not working, and the rest of the period is determined to be OLF (code of 5).

The second condition in the second path tests to see if the value in the WEEKLY LABOR STATUS array is not a code of 4; if COD is a job number then the job number is placed into WEEKLY LABOR STATUS. If there are hours for the week and if the respondent was not working for the employer during this week, then the hours for the week are set to zero if HOURS WORKED is greater than zero. Otherwise, HOURS WORKED receives whatever value is in HOURSWEEK.

The third path FILL can take is if this is a period not working and if there is a dual job. Then, the job number is deleted from DUAL JOBS and HOURSWEEK for the job are subtracted from HOURS WORKED.

Finally, if there are more than four dual jobs in DUA LJOBS then no other job numbers are added to DUAL JOBS because the array for each week is limited to four dual job variables.

User Notes: A few last notes about FILL:

1. Civilian work takes precedence over any other activity. If the respondent has a civilian job while in the military, then the civilian job code replaces the military code in the WEEKLY LABOR STATUS array.
2.
 1. The order of precedence in the construction of the WEEKLY LABOR STATUS array after a civilian job is as follows:
 - a. a code of 3, associated with an employer but periods not working with employer are missing; if any period not working is missing, then the entire period of the job is assigned a 3. In 1979, there are 274 cases with invalid period dates, and in each of the following survey years, there are fewer than 60 cases
 - b. a code of 4, unemployed
 - c. a code of 5, OLF
 - d. a code of 2, not working but OLF vs unemployed status is unknown
 - e. a code of 7, active military service
 - f. a code of 0, no information is reported to account for the week
 3. About 32 cases have a week in which JOB # 1 from a survey week first appears in the DUAL JOBS array rather than the WEEKLY LABOR STATUS array. This occurs when (1) there is a discrepancy between the date of the previous interview date as it appears on the info sheet that the interviewer uses at the time of the interview and the interview date recorded at the previous interview or (2) the starting date and ending date for a job across interview years are the same due primarily to the way the dates are floored and ceiled. In all these cases, an erroneous entry appears in the DUAL JOBS array for that given week.

Changes between the 1979–86 and the 1979–87 Work History Data File

In 1987, a few changes were made to the program that created the work history data file. These changes from the 1986 program affected the created labor force participation key variables, the STATUS array, and the HOUR array. In addition, two sets of variables were added for each year: (1) WHYLEFT, the reason that the respondent left each job for each year if they were not currently working at that job at the

date of the interview and (2) BREASON, the reason that the respondent was not looking for work during each of the possible six periods not working between jobs for each year.

The following is a more detailed discussion of the changes in the code that were made and the effects of those changes on the key variables and the week-by-week arrays:

1. In the CALC PROC, the stop dates for all jobs and all periods not working were set to the interview date if the dates were greater than zero and if they were greater than the interview date. These changes resulted in an increased number of weeks unaccounted for in calculating weeks not working and in changes in the number of weeks unemployed and out of the labor force across all of the key variables for each survey year. Most of the changes were a difference of one week or a change to an invalid value.
2. In the FILL PROC when hours were subtracted from the weekly HOURS WORKED array, a check was made to determine if the subtraction resulted in a value greater than or equal to zero. If it did not, the value in the HOURS WORKED array for that week was set to zero. If there was no dual job for that week, then the value in the HOURS WORKED array for that week was set to zero; previously, a subtraction was performed.
3. Before these changes, some cases had negative hours (not including missing value codes) in some of the weeks in the HOURS WORKED array. Now, all of the values in the HOURS WORKED array are positive except for the standard missing values. These changes resulted in an overall decrease in the number of hours reported in a given week and in the number of hours calculated for the last calendar year and since the date of the last interview for those cases that were affected.

The following table lists the key variables for each year that had a change in values and the number of cases that had a change in the calculation of that key variable between the 1979–87 work history creations and the previous years.

TABLE 1: Cases Changed by 1987 Work History Program

# of Cases Changed	Reference Number	Variable Name	Year	Variable Title
3	R04071.10	MILWKSC	1980	Number of Weeks Service in Active Armed Forces in Past Calendar Year
2	R06457.10	MILWKSC	1981	Number of Weeks Service in Active Armed Forces in Past Calendar Year
1	R08977.10	MILWKSC	1982	Number of Weeks Service in Active Armed Forces in Past Calendar Year
7	R02157.10	HOURC	1979	Number of Hours Worked in Past Calendar Year *KEY*
4	R02157.	WORKC	1979	Number of Weeks Worked in Past Calendar Year *KEY*
102	R02158.	WUMPC	1979	Number of Weeks Unemployed in Past Calendar Year *KEY*
102	R02159.	WOLFC	1979	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
7	R02157.01	MISSC	1979	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year
6	R02153.10	HOURL	1979	Number of Hours Worked since Last Int *KEY*
4	R02153.	WORKL	1979	Number of Weeks Worked since Last Int *KEY*

TABLE 1: Cases Changed by 1987 Work History Program

# of Cases Changed	Reference Number	Variable Name	Year	Variable Title
5	R02154.	WUMPL	1979	Number of Weeks Unemployed since Last Int *KEY*
5	R02155.	WOLFL	1979	Number of Weeks out of Labor Force since Last Int *KEY*
8	R02153.01	MISSL	1979	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
89	R04073.	HOURC	1980	Number of Hours Worked in Past Calendar Year *KEY*
62	R04072.	WORKC	1980	Number of Weeks Worked in Past Calendar Year *KEY*
345	R04074.	WUMPC	1980	Number of Weeks Unemployed in Past Calendar Year *KEY*
365	R04075.	WOLFC	1980	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
117	R04072.01	MISSC	1980	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year
102	R04068.	HOURL	1980	Number of Hours Worked since Last Int *KEY*
71	R04067.	WORKL	1980	Number of Weeks Worked since Last Int *KEY*
180	R04069.	WUMPL	1980	Number of Weeks Unemployed since Last Int *KEY*
211	R04070.	WOLFL	1980	Number of Weeks out of Labor Force since Last Int *KEY*
110	R04067.01	MISSL	1980	Number of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
45	R06466.	HOURC	1981	Number of Hours Worked in Past Calendar Year *KEY*
30	R06463.	WORKC	1981	Number of Weeks Worked in Past Calendar Year *KEY*
412	R06464.	WUMPC	1981	Number of Weeks Unemployed in Past Calendar Year *KEY*
432	R06465.	WOLFC	1981	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
44	R06463.01	MISSC	1981	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year
47	R06462.	HOURL	1981	Number of Hours Worked since Last Int *KEY*
31	R06458.	WORKL	1981	Number of Weeks Worked since Last Int *KEY*
243	R06459.	WUMPL	1981	Number of Weeks Unemployed since Last Int *KEY*
265	R06460.	WOLFL	1981	Number of Weeks out of Labor Force since Last Int *KEY*
44	R06458.01	MISSL	1981	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
46	R08968.	HOURC	1982	Number of Hours Worked in Pastcalendar Year *KEY*

TABLE 1: Cases Changed by 1987 Work History Program

# of Cases Changed	Reference Number	Variable Name	Year	Variable Title
35	R08969.	WORKC	1982	Number of Weeks Worked in Past Calendar Year *KEY*
485	R08970.	WUMPC	1982	Number of Weeks Unemployed in Past Calendar Year *KEY*
493	R08971.	WOLFC	1982	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
52	R08969.01	MISSC	1982	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year
50	R08972.	HOURL	1982	Number of Hours Worked since Last Int *KEY*
37	R08973.	WORKL	1982	Number of Weeks Worked since Last Int *KEY*
306	R08974.	WUMPL	1982	Number of Weeks Unemployed since Last Int *KEY*
318	R08975.	WOLFL	1982	Number of Weeks out of Labor Force since Last Int *KEY*
53	R08973.01	MISSL	1982	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
37	R11452.	HOURC	1983	Number of Hours Worked in Past Calendar Year *KEY*
26	R11453.	WORKC	1983	Number of Weeks Worked in Past Calendar Year *KEY*
537	R11454.	WUMPC	1983	Number of Weeks Unemployed in Past Calendar Year *KEY*
535	R11455.	WOLFC	1983	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
68	R11456.	MISSC	1983	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar
38	R11457.	HOURL	1983	Number of Hours Worked since Last Int *KEY*
28	R11458.	WORKL	1983	Number of Weeks Worked since Last Int *KEY*
370	R11459.	WUMPL	1983	Number of Weeks Unemployed since Last Int *KEY*
369	R11460.	WOLFL	1983	Number of Weeks out of Labor Force since Last Int *KEY*
68	R11461.	MISSL	1983	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
33	R15204.	HOURC	1984	Number of Hours Worked in Past Calendar Year *KEY*
20	R15205.	WORKC	1984	Number of Weeks Worked in Past Calendar Year *KEY*
479	R15206.	WUMPC	1984	Number of Weeks Unemployed in Past Calendar Year *KEY*
480	R15207.	WOLFC	1984	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
47	R15208.	MISSC	1984	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year

TABLE 1: Cases Changed by 1987 Work History Program

# of Cases Changed	Reference Number	Variable Name	Year	Variable Title
34	R15209.	HOURL	1984	Number of Hours Worked since Last Int *KEY*
20	R15210.	WORKL	1984	Number of Weeks Worked since Last Int *KEY*
314	R15211.	WUMPL	1984	Number of Weeks Unemployed since Last Int *KEY*
315	R15212.	WOLFL	1984	Number of Weeks out of Labor Force since Last Int *KEY*
47	R15213.	MISSL	1984	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
37	R18911.	HOURC	1985	Number of Hours Worked in Past Calendar Year *KEY*
24	R18912.	WORKC	1985	Number of Weeks Worked in Past Calendar Year *KEY*
408	R18913.	WUMPC	1985	Number of Weeks Unemployed in Past Calendar Year *KEY*
412	R18914.	WOLFC	1985	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
45	R18915.	MISSC	1985	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year
37	R18916.	HOURL	1985	Number of Hours Worked since Last Int *KEY*
24	R18917.	WORKL	1985	Number of Weeks Worked since Last Int *KEY*
275	R18918.	WUMPL	1985	Number of Weeks Unemployed since Last Int *KEY*
279	R18919.	WOLFL	1985	Number of Weeks out of Labor Force since Last Int *KEY*
46	R18920.	MISSL	1985	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int
35	R22582.	HOURC	1986	Number of Hours Worked in Past Calendar Year *KEY*
20	R22583.	WORKC	1986	Number of Weeks Worked in Past Calendar Year *KEY*
300	R22584.	WUMPC	1986	Number of Weeks Unemployed in Past Calendar Year *KEY*
301	R22585.	WOLFC	1986	Number of Weeks out of Labor Force in Past Calendar Year *KEY*
30	R22586.	MISSC	1986	Percent of Weeks Unaccounted for in Calculating Weeks Worked in Past Calendar Year
35	R22587.	HOURL	1986	Number of Hours Worked since Last Int *KEY*
20	R22588.	WORKL	1986	Number of Weeks Worked since Last Int *KEY*
229	R22589.	WUMPL	1986	Number of Weeks Unemployed since Last Int *KEY*
230	R22590.	WOLFL	1986	Number of Weeks out of Labor Force since Last Int *KEY*
30	R22591.	MISSL	1986	Percent of Weeks Unaccounted for in Calculating Weeks Worked since Last Int

Changes between the 1979–87 and the 1979–88 Work History Data File

Most changes made to the work history program between the 1979–87 and 1979–88 data files did not affect the content of the variables themselves. Some changes were made to simplify the reading and use of the program in the future. Format changes were also made to allow for larger variable lengths. Because 1988 is the 10th year of the NLSY79, variables such as a job number, which provided only one space for the survey year, were expanded. The DUAL JOB array was no longer concatenated. Instead, four variables are present for each week, allowing (as before) for up to four dual jobs per week.

Substantive changes are not major and are a function of changes in the questionnaire:

1. GOVTJOB in 1988 is set to valid missing for all respondents. This question was dropped from the survey.
2. HOURSWEEK in 1988 also includes additional hours worked at home if any are reported. The 1988 questionnaire asked respondents separately about hours worked at home for a job. If any hours worked at home were reported, respondents were asked if their total hours worked per week included those hours worked at home. If not, the total hours worked per week and the hours worked at home were added together to get a total number of hours worked per week anywhere for a job.

Changes between the 1979–88 and the 1979–89 Work History Data File

Several additions were made to the variable structure for the 1979–89 work history data file. These changes did not affect the content or substance of already existing variables.

1. A JOBSEVER variable was created for each year from 1979–89. This variable is a cumulative count of the number of different jobs that have ever been reported by a respondent up to the date of interview for the survey year. Users should note that, as with calculations for the TENURE variables discussed earlier in this program description, employers can only be traced through contiguous years. In non-contiguous years, the number of jobs reported may be slightly inflated in some cases.
2. SEX and RACE variables have been added to the work history stratifications by sex and race for all respondents. The SAMPLE_ID variable remains in the dataset for those requiring further sample delineation.

Some data changes have been made in existing variables as well. Two of these reflect corrections that have been made in the calculations for 1987 and 1988 variables.

1. The WEIGHT variable for each year has been recoded for non-interview cases from -5 to 0. This recoding makes these weight variables consistent with those found in the main data set.
2. The 1987–88 TENURE variables were in error in the 1979–87 and 1979–88 work history releases. An error in the program statements which calculate this variable resulted in large numbers of respondents with valid values receiving -3 values instead. This error has been corrected and the changes have been incorporated in the current release.
3. In the 1979–88 work history release, the HOURSWEEK variable was to include additional hours worked at home on a job, if reported. Although this was true for JOBS #6-10, the necessary programming changes for JOBS #1-5 were inadvertently omitted from the program. Therefore, JOBS #1-5 were calculated as they have been in previous work history programs, based upon one question without qualification for any additional hours worked at home. The omission has been corrected and the changes have been incorporated in the current release.
4. In 1988, 116 cases reported a 3rd within-job gap for at least one job. The information for these gaps was erroneously included as information for a 4th within-job gap. The 3rd within-job gaps for

these cases would have been missing. This has been corrected in the 1979–89 release. Additionally, information on a 4th within-job gap for at least one job has been included for 18 cases.

Changes between the 1979–89 and the 1979–90 Work History Data File

A minor modification was made to the HRP PROC (the procedure at the end of each year's program which calculates HOURLYWAGE from PAYRATE and TIMERATE). Any PAYRATE which has a value of 9999995 is now set to -4 by the HRP PROC. This 9999995 value indicates a case for which the dollars and cents PAYRATE exceeded \$100,000.00.

Some data updates were made to existing variables as well. Users have already been notified of the erroneous data for affected cases with the release of the 1989 main NLSY79 and work history data files, and in the Summer 1991 (No. 68) issue of NLSUPDATE.

1. Specific job information for 70 cases was edited, for one or more jobs, due to improper identification of CPS jobs in the Employer Supplements.
2. Corrections were made to 23 cases for 1988 PAYRATES and/or 1988–89 HOURLYWAGES. These cases exceeded \$100,000.00 and should have been assigned the 9999995 value. While some contained that value, some retained an erroneous dollars and cents value in PAYRATE. In either case, the HOURLYWAGEs were calculated based upon an incorrect PAYRATE figure. The above-mentioned adjustment to HRP PROC will prevent the calculation of HOURLYWAGE figures from the truncated 9999995 value in the future.

Changes between the 1979–91 and the 1979–92 Work History Data File

A change has been made to the structure of the 1979–92 work history data file on magnetic tape only. Due to the volume of the current work history data file, the data were split into two records. The first record contains the data for the STATUS, HOURS and DUALJOBS arrays. The second contains the remainder of the data, pertaining to specific job characteristics, gaps in employment and summary labor force activity variables. Those wishing to use only job specific variables can now do so without being required to process information for an entire case to do so. Those wishing to incorporate the arrays in analysis can access them in a separate record. Tape users should refer to the record layout and format table provided in this package of documentation for details on the exact location of each variable. This change does not affect the content or substance of already existing variables.

A correction was made to an existing set of variables as well. Users have already been notified of the inadvertent omission of hourly rates of pay for those respondent reporting earnings on a semi-monthly basis in the 1990 and 1991 main NLSY79 and work history data files, and in the Winter 1993 (No. 74) issue of NLSUPDATE. These cases have been corrected in the 1979–92 version of the work history data file.

Changes between the 1979–92 and the 1979–93 Work History Program

Changes occurring between these releases were not substantive. Adjustments made for both the size of the data file and to accommodate the CAPI-collected data were necessitated in the work history program and will only be noticeable to those reviewing the actual 1993 work history program, included in the 1979–93 work history documentation. Users examining the program will find three types of changes:

1. Due to the size of the work history data file, it is prohibitive to maintain an in-house CHRR data file with 10 jobs, from which the five-job public tape is then produced. Instead, the previous year's public release, containing data on five jobs only, is used to create the base data file for past

survey years. The current survey data is appended to the 5-job file. A 10-job working data file is then created in the course of the work history program, ONLY for the current year's data, to allow created variables being calculated for the current survey year to incorporate data from up to 10 jobs. Detailed information on only the first five jobs is retained for the current release.

2. The work history program previously required a separate data set for data on additional job gaps (beyond the first three within-job gaps and first four between-job gaps). This extra data file is no longer necessary. The CAPI data file contains all data (except weights, which are added from a different source), necessary to produce the work history data file.
3. Beginning with the 1979–93 release, the formats for the PAYRATE variables have been extended to 8 characters to accommodate values up to 99999999 (\$999999.99). Previously, these cases containing these variables had been assigned a PAYRATE value of 9999995 and set to -4 in the HRP PROC, which creates the HOURLYWAGE variables. Valid PAYRATE and HOURLYWAGE values are now present in these cases.

Changes between the 1979–93 and the 1979–94 Work History Data File

The recall experiment (an experiment to test the recall of respondents over a two year period) was conducted with over 850 randomly selected respondents during the 1994 interview. For this experiment, respondents were treated as if the 1993 interview never took place; the interview was conducted as if the 1992 interview was the most recent. Because data from 1993 were already incorporated into the work history data file, we sought to keep redundant data from the 1994 interview for the "recall" cases from overwriting the already incorporated 1993 data. Efforts were made to eliminate the overlap between the information reported in 1993 and 1994 for those cases, and to keep only the information from the 1994 interview that covered the period since the 1993 interview. However, there were isolated circumstances in which this was not possible. These relate to the assignment of "OLF" versus "unemployed" labor force status during periods not working which contain both types of statuses (see earlier discussion in this document). While it is possible to determine which part of a period not working occurred since the 1993 interview, it is not possible to make the same determination for "OLF" versus "unemployed" status during those periods. Therefore, it is likely that in some cases these statuses would not have been assigned correctly to certain periods not working. See Appendix 16 in this codebook supplement document for further details on the recall experiment.

Changes between the 1979–94 and the 1979–96 Work History Data File

Through survey year 1994, the work history data file was created by running PL/I programs on an IBM mainframe. In 1996, the volume of the work history data file dictated a change to a more efficient method of production. To create the 1979–96 data file, the PL/I program was converted to SQL code. Relevant variables from the main NLSY79 data file were loaded into a relational data base, from which the work history data file was generated. The SQL code that generated the data file replicates the PL/I program, both in substance and function.

No revisions were made to the 1979–94 job-specific data created by the PL/I programs in past rounds. For respondents with missing interviews between the last interview and 1996, the STATUS, HOUR and DUAL JOB arrays were updated by the SQL program in the same manner as in past years with the PL/I programs.

Although the SQL programs are not included in this appendix, the separate addendum contains the PL/I programs from past years. A list of the main NLSY79 variables used in the creation of the 1979–96 work history data set is included at the end of this appendix.

Changes between the 1979–96 and the 1979–98 Work History Data File

For the first time with the 1979–98 work history release, Windows-based extraction software accompanied the data file.

Users should be aware that the TENURE variable for job #2, reported in 1980, was found to be in error on the 1979–96 work history release only. This variable has been replaced with the correct data on the 1979–98 work history release.

Changes between the 1979–98 and the 1979–2000 Work History Data

The round 19 combined main youth-work history release marks the first time that the work history data are being released in combination with the main NLSY79 data. Data items formerly available only in a separate work history data file, including the week-by-week arrays, are now available in a series of new areas of interest on the public release CD, using the same extraction software as the main NLSY79 data. This eliminates the need for multiple extracts and merging of data from different CDs, as well as the duplication of some information specific to individual jobs and respondents between data files.

DESCRIPTION AND CODES FOR VARIABLES IN 1979–2000 NLSY79 WORK HISTORY DATA

A significant change has been implemented with respect to the 1979–2000 work history data. For the first time on the 2000 (round 19) public release CD, the work history file has been combined with the other NLSY79 data. This eliminates the need for multiple extracts and merging of data from different CDs, as well as the duplication of some information specific to individual jobs and respondents. Below is a listing of items which formerly comprised the separate work history data file and their disposition on the combined NLSY79 2000 CD. Variable coding information, as well as formulas for combining job-specific characteristics from several sources, are included where relevant.

Work history weekly array variables

The foundation of the work history data file is the set of week-by-week arrays depicting labor force status, total number of hours, and dual job holdings if any, for each week since January 1, 1978. These array variables are now found in three new areas of interest on the combined main–work history NLSY79 release. The construction and coding for each of the three arrays are described below, listed by their new area of interest.

Area of Interest: WORK HISTORY-WEEKLY LABOR STATUS

The WEEKLY LABOR STATUS array is the work history week array. Each variable corresponds to a week relative to 1/1/78.¹ There are 1,201 variables in the 1979–2000 WEEKLY LABOR STATUS array—one for each of the 1,201 weeks from 1/1/78 to 12/31/00.¹ There are no missing data codes, and the codes that are in the array are as follows:

- 0 = no information reported to account for week.
- 2 = not working (unemployment vs. out of the labor force cannot be determined.)
- 3 = associated with an employer but the periods not working for the employer are missing. If all of the time with the employer cannot be accounted for, a 3 is loaded into the STATUS array instead of a job code.
- 4 = unemployed. If a respondent is not working and part of the time is spent looking for work or on layoff, the exact weeks spent looking for work is unknown. As a result, the number of weeks spent looking is assigned to the middle part of the period not working.
- 5 = out of the labor force.
- 7 = active military service. If a respondent has a civilian job while in active military service, the civilian job code is loaded into the array instead of a code of 7.
- 100 = worked. The code represents the appropriate work history year multiplied by 100 plus the job number for that employer in that year. For example, 102=year 1, job 2; 305=year 3, job 5. This allows one to associate any characteristic for a job with that week. If a respondent has more than one job at the same time, the job number that is loaded into the array is determined by the starting date of the job with the lowest job number, not by any particular characteristics of the job such as the number of hours worked at the job. The year in the job code is the year in which the job is reported. Jobs held in year 2, but reported in year 10 would be assigned job numbers beginning with 1001 instead of 201.

¹ All week number references in this program are relative to 1/1/78 and end with the most recent interview date. A week #0 is included at the beginning of the week-by-week array structures to indicate time prior to 1/1/78. Users are discouraged from incorporating data contained in this week in analysis. Researchers should instead use information from the 1979 interview concerning labor force activity prior to 1/1/78 in order to construct event histories of a more thorough nature. (Some information concerning labor force activity for respondents prior to the time frame of the initial interview is asked on an age restricted basis for respondents still in their teens at the time of interview.)

User Notes: In some cases, a respondent reports a period not working that is part OLF and part unemployed. In these cases, a week-specific distinction between OLF and unemployed cannot be made. Users should refer to the Work History Program Description in this appendix for a discussion of how OLF and unemployed codes are assigned to the WEEKLY LABOR STATUS array in the event that such a period occurs.

Area of Interest: WORK HISTORY-HOURS WORKED

The HOURS WORKED array contains the usual hours worked per week at all jobs. There are 1,201 variables in the 1979–2000 HOURS WORKED array—one for each of the 1,201 weeks from 1/1/78 to 12/31/00.² The codes are as follows:

- 1-95 = usual hours worked per week
- 96 = 96 or more hours per week
- 5 = noninterview
- 4 = valid skip
- 3 = invalid skip
- 2 = don't know
- 1 = refusal

User Notes: Beginning in 1993, the first all-CAPI survey year, the maximum hours allowed per week is 168.

Area of Interest: WORK HISTORY-DUAL JOBS

The DUAL JOBS array contains job numbers for any weeks when the respondent worked at more than one job. There are 3,610 variables in the DUAL JOBS array – up to four for each of the 1,201 weeks from 1/1/78 to 12/31/00.² DUAL JOBS array variables are present if a dual job was reported. The variables are written in the following order (by week and then by job)³:

- Week 0 < 1/1/78, Job # 1
- Week 0, Job # 2
- Week 0, Job # 3
- Week 0, Job # 4
- Week 1, Job # 1
- Week 1, Job # 2
- Week 1, Job # 3
- Week 1, Job # 4
- etc.

The codes are as follows:

- 0 = no dual job
- >100 = dual job year and job number

For example, if a respondent worked at three jobs at the same time, the code for the lowest job number would be in the WEEKLY LABOR STATUS array, and the codes for the other two jobs would be in the DUAL JOBS array (see item 3 in the user notes below). If the three jobs that the respondent held during week 190 from the 1981 survey were jobs 1, 5, and 6, then WEEKLY LABOR STATUS would contain

² See footnote 1.

³ All variables have standard missing value codes unless otherwise noted.

the value '301' for that week, and two DUAL JOBS array variables for week 190 would contain the values '305' and '306'.

User Notes: A few additional notes are in order:

1. The maximum number of job codes allowed in DUAL JOBS is 4. The variable descriptions for variables in the DUAL JOBS (WORK HISTORY) area of interest indicate the relevant job number and week.
2. The DUAL JOBS array does not provide labor force status in the detailed manner of the WEEKLY LABOR STATUS array. It contains only second, third and fourth job numbers for weeks in which the respondent reports more than one employer.
3. Users should be aware that it is possible in survey years 1979-92 for the CPS job number to appear in the DUAL JOBS array instead of the WEEKLY LABOR STATUS array, as would be expected. In most cases, the CPS job will be the lowest number job for a given year. However, this is not always the case. Each year contains a relatively small number of cases for which JOB #1 is not the CPS job. For these cases, the job number assigned by the work history program will not necessarily be the lowest one for that year. In cases for which the CPS job is not held simultaneously to any other job, the job number for the CPS job will appear in the WEEKLY LABOR STATUS array as expected. However, in cases for which the CPS job is held simultaneously with another job with a lower job number, the possibility exists that the job number for the CPS job will appear in the DUAL JOBS array instead of the WEEKLY LABOR STATUS array. Mechanical changes implemented in the 1993 CAPI instrument to ensure that the CPS job is always the first job should prevent this possibility from 1993 forward.

Non-array work history variables

The variables listed below have traditionally been included on the work history data file only. On the combined main-work history release, these variables are now contained in one of several new work history areas of interest. They are listed below by new area of interest. Variables marked with an asterisk (*) contain an actual consecutive week number, ranging from week number 0-1200, with the week of January 1, 1978, being week #1. Week #0 represents information for time prior to that date.

Area of Interest: WORK HISTORY-HISTORY

LASTINT*	Week of last interview
INT*	Week of current interview

Area of Interest: WORK HISTORY-CALENDAR YEAR

NUMBER	Job number that is loaded into the STATUS array for each job. The 1 st two digits of the number are the year (01 thru 18) and the 2 nd two digits are the job for that year (job 01 thru 10)
CAL_YEAR_JOBS#	Number of jobs in past calendar year
NWMISSC	Percentage of weeks not employed in past calendar year that cannot be split between unemployed and out of the labor force

Area of Interest: WORK HISTORY-JOBS

START*	Starting week of each job
STOP*	Stopping week of each job
PERIODSTART*	Starting week of each period not working for each job
PERIODSTOP*	Stopping week of each period not working for each job

Area of Interest: WORK HISTORY-GAPS BETWEEN JOBS

BSTART* Week started each period not working between jobs
BSTOP* Week stopped each period not working between jobs.

Area of Interest: WORK HISTORY-SINCE LAST INTERVIEW

LASTINT_JOBS Number of jobs since the date of the last interview
NWMISSL Percentage of weeks not employed since the date of the last interview that cannot be split between unemployed and out of the labor force

Area of Interest: WORK HISTORY-MILITARY

MSTART1* Starting week of first period of active military service.
MSTART2* Starting week of second period of active military service.
MSTOP1* Stopping week of first period of active military service.
MSTOP2* Stopping week of second period of active military service.

Variables created by the work history programs included in NLSY79 main data file

The work history programs produced a set of variables each year that were included both on the separate work history data file and the NLSY79 main data file. In the combined main-work history data release, these items are found in their traditional place among public release data items. They are listed below, with an example reference number for the most current year for each variable in parentheses.

Job variables (data present for up to 5 jobs for each survey year)

TENURE (R70052.) Total weeks tenure at each job as of interview date
HOURLYWAGE (R70057.) Usual wage earned at each job converted to an hourly rate

Survey year variables, since date of last interview (data present for each survey year)

MILWKSL (R70087.) Weeks of active military service since date of last interview
WORKL (R70082.) Number of weeks worked since date of last interview
HOURL (R70081.) Number of hours worked since date of last interview
WUMPL (R70083.) Number of weeks unemployed since date of last interview
WOLFL (R70084.) Number of weeks out of the labor force since date of last interview
MISSL (R70085.) Percentage of weeks unaccounted for in calculating weeks worked since date of last interview

Survey year variables, calendar year prior to survey year (data present for each survey year)

MILWKSC (R70088.) Weeks of active military service in past calendar year
WORKC (R70077.) Number of weeks worked in past calendar year
HOURC (R70076.) Number of hours worked in past calendar year
WUMPC (R70078.) Number of weeks unemployed in past calendar year
WOLFC (R70079.) Number of weeks out of the labor force in past calendar year
MISSC (R70080.) Percentage of weeks unaccounted for in calculating weeks worked in past calendar year

Survey year variables (data present for each survey year)

WBID (R70086.) Number of weeks since date of last interview
JOBSEVER (R70093.) Number of jobs ever reported as of interview date

Variables picked up from NLSY79 main data file

The separate work history data file contained a large number of duplicated items pertaining to jobs, job gaps, and the individual respondents that were copied straight from the main data file. On the combined main-work history data release, these items can be found in their traditional place among public release data items. They are listed below by type of variable. Example reference numbers for the most current year for each variable are listed in parentheses.

Respondent variables

PUBLIC ID	(R00001.)	Respondent's public identification code
SEX	(R02148.)	Respondent's sex
RACE	(R02147.)	Respondent's race
SAMPLE_ID	(R01736.)	Respondent's sample type
BIRTHM_79	(R00003.)	Respondent's month of birth from the 1979 interview.
BIRTHD_79	(R00004.)	Respondent's day of birth from the 1979 interview.
BIRTHY_79	(R00005.)	Respondent's year of birth from the 1979 interview.
BIRTHM_81	(R04101.)	Respondent's month of birth from 1981 interview or from 1979 interview if 1981 non-interview
BIRTHY_81	(R04103.)	Respondent's year of birth from 1981 interview or from 1979 interview if 1981 non-interview

Survey year variables (data present for each survey year)

WEIGHT	(R70062.)	Sampling weight
INTM	(R69633.)	Month of interview
INTD	(R69633.01)	Day of interview
INTY	(R69633.02)	Year of interview

Job variables (data present for up to 5 jobs for each survey year)

STARTM	(R70009.)	Starting month of job
STARTD	(R70009.01)	Starting day of job
STARTY	(R70009.02)	Starting year of job
STOPM	(R69999.)	Stopping month of job
STOPD	(R69999.01)	Stopping day of job
STOPY	(R69999.02)	Stopping year of job
PAST*	(R65532., R65537.)	Starting date of each job is before, the same as, or after the date of the last interview? *Combination of two variables in CAPI interviews
CURRENT	(R65550.)	Currently working for employer at date of interview
WHYLEFT*	(R65555.)	Reason left job *Please note that coding varies over time
CPSJOB*	(R70019.)	Is employer the CPS employer? In other words, is employer the current or most recent employer? *This variable is all -4's for 1979, when job 1 is the CPS job. Beginning in 1993, job #1 is always the CPS job if there is one.
OCCUPATION	(R65918.)	Kind of work usually done for employer - 1970 codes
INDUSTRY	(R65913.)	Kind of business or industry of employer - 1970 codes
CLASSWORKER*	(R65923.)	Employee of a private company, a government employee, self-employed, or working without pay at a family business or farm of worker at each job? *Please note coding changes beginning in 1994
HOURDAY	(R65777.)	Hours per day usually worked at job
PAYRATE	(R65935.)	How much usually earned at job
TIMERATE	(R65940.)	Payrate reported per hour, per day, per week, or what at job
UNION*	(R66899.)	Wages or salary at each job set by a collective bargaining agreement between employer and a union or employee association? *Please note restriction on hours per week working varies over time

Appendix 18: Work History Data

GOVTJOB*	(R23786.)	Is one or more of government job codes circled on the cover of employer supplement? In other words, is job a government-sponsored job? *Question eliminated in 1988
WEEKSNOT WORKED	(R65610.)	Between start and stop dates for each job for each year, were there any periods of a full week or more during which R did not work for employer, not counting paid vacations or paid sick leave?
PRETEN	(R65547.)	The total number of months worked for employer before date of last interview
<i>Gaps within jobs (data present for up to 4 gaps within each job for each survey year)</i>		
REASON	(R65654.)	Reason not working for employer for each period not working
ALL	(R65691.)	How many weeks looking for work or on layoff during each period not working?
LOOK	(R65719.)	Number of weeks not working and looking for work or on layoff during each period not working
<i>Gaps between jobs (data present for up to 6 gaps between jobs for each survey year)</i>		
BALL	(R67009.)	How many weeks not working spent looking for work or on layoff during periods not working between jobs?
BLOOK	(R67017.)	Number of weeks looking for work or on layoff during periods not working between jobs
BREASON	(R67025.)	Reason not looking for work during periods not working between jobs

Variables compiled from NLSY79 main data variables

Some variables in the separate work history data files were created from multiple variables in the main data file. They are listed below, along with the items that were used to compile them in the most recent year available.

HOURSWEEK*	(R65782., R65797.)	Hours per week usually worked at each job *Beginning in 1988 this variable includes additional hours worked at home if any are reported
PREVIOUSEMP#*	(R65527.)	Job number assigned to employer from last interview as listed on information sheet for each interview year. This allows for matching employers between consecutive interview years only. *In pre-1993 years, this variable was combined from two separate items listing id numbers for different sets of jobs. See Appendix 9, "Linking Employers Through Survey Years," for a more detailed discussion.

WORK HISTORY PROGRAMS

The PL/1 program used to create the work history variables through 1994 is available to researchers. This program can be examined by those who desire details about the creation of the work history variables not included elsewhere in this appendix. Due to the length of this program, it is not printed in this document. This program is available only in electronic format. Users interested in obtaining this "Addendum to Appendix 18" should contact NLS User Services (see the cover of this codebook supplement for contact information).

VARIABLES USED IN CREATION OF 1996 AND SUBSEQUENT WORK HISTORY DATA FILES

Beginning in 1996, the work history variables were created using a new SQL program. The SQL program, which mirrors the older PL/1 program, is not available to users. However, the following pages list the variables used as inputs to the SQL program. Users who need more information should contact NLS User Services.

Users should be aware that not all of variables listed below appear in the main NLSY79 data file. Variables with no valid data for any respondent, jobs 6-10, within-job gap 4 and between-job gaps 5-6 are not currently included in the main file.

1996 Work History Input Variables

/* The following variables were loaded into a relational data base and used as direct input for creation */
/* of the 1979-96 Work History data file.*/

CASEID	These variables repeat for employers 1-10:		
SAMPWT96	EMP1PREVID to	E1_33_3 to E10_33_3	E1_74D to E10_74D
LINTDATE	EMP10PREVID	E1_34_3 to E10_34_3	E1_74E to E10_74E
Q_1C	E1_4B to E10_4B	E1_36_3 to E10_36_3	E1_74K to E10_74K
Q4_1A, 1B	E1_6 to E10_6	E1_40_3 to E10_40_3	E1_74M to E10_74M
Q4_5A	E1_8 to E10_8	E1_33_4 to E10_33_4	E1_74Q to E10_74Q
Q4_6A	E1_8A to E10_8A	E1_34_4 to E10_34_4	E1_74R to E10_74R
Q4_9, 9A, 9A1, 9B, 9B1	E1_23 to E10_23	E1_36_4 to E10_36_4	E1_74U to E10_74U
Q4_10	E1_23A to E10_23A	E1_40_4 to E10_40_4	E1_74V to E10_74V
Q4_11, 11B	E1_26 to E10_26	E1_51 to E10_51	E1_75B to E10_75B
Q4_12, 12A, 12B, 12C	E1_28 to E10_28	E1_52A to E10_52A	E1_75D to E10_75D
Q4_13, 13A	E1_30_1 to E10_30_1	E1_52D to E10_52D	E1_75G to E10_75G
Q4_30	E1_31_1 to E10_31_1	E1_55Dc to E10_55Dc	E1_75H to E10_75H
OMILCODE	E1_30_2 to E10_30_2	E1_55F to E10_55F	E1_75J to E10_75J
	E1_31_2 to E10_31_2	E1_55I to E10_55I	E1_75K to E10_75K
These variables repeat for gaps 1-6:	E1_30_3 to E10_30_3	E1_56A to E10_56A	E1_75Q to E10_75Q
Q7_10_1 to Q7_10_6	E1_31_3 to E10_31_3	E1_56B to E10_56B	E1_75S to E10_75S
Q7_11_1 to Q7_11_6	E1_30_4 to E10_30_4	E1_56C to E10_56C	E1_75V to E10_75V
Q7_12_1 to Q7_12_6	E1_31_4 to E10_31_4	E1_56Ka to E10_56Ka	E1_75W to E10_75W
Q7_16_1 to Q7_16_6	E1_33_1 to E10_33_1	E1_56Kb to E10_56Kb	E1_75Y to E10_75Y
Q7_19_1 to Q7_19_6	E1_34_1 to E10_34_1	E1_56Kc to E10_56Kc	E1_75Z to E10_75Z
	E1_36_1 to E10_36_1	E1_71A to E10_71A	E1_76F to E10_76F
	E1_40_1 to E10_40_1	E1_71I to E10_71I	E1_76H to E10_76H
	E1_33_2 to E10_33_2	E1_71J to E10_71J	E1_76K to E10_76K
	E1_34_2 to E10_34_2	E1_71P to E10_71P	E1_76L to E10_76L
	E1_36_2 to E10_36_2	E1_71R to E10_71R	E1_88B to E10_88B
	E1_40_2 to E10_40_2	E1_73J to E10_73J	

1998 Work History Input Variables

/* The following variables were loaded into a relational data base and used as direct input for creation */
/* of the 1979–98 work history data file.*/

PUBLIC_ID	These variables repeat for gaps 1-6:
SAMPWEIGHT	Q7-10.01-D to Q7-10.06-D
LINTDATE	Q7-10.01-M to Q7-10.06-M
CURDATE	Q7-10.01-Y to Q7-10.06-Y
Q4-1A, 1B	Q7-11.01-D to Q7-11.06-D
Q4-5A	Q7-11.01-M to Q7-11.06-M
Q4-6A-D, ~M, ~Y	Q7-11.01-Y to Q7-11.06-Y
Q4-9, 9A	Q7-12.01 to Q7-12.06
Q4-10	Q7-15.01 to Q7-15.06
Q4-11, Q4-11B-D, ~M, ~Y	Q7-16.01 to Q7-16.06
Q4-30	Q7-17.01 to Q7-17.06
Q5-JUMP	Q7-19.01 to Q7-19.06

These variables repeat for employers 1-10:

QES-4B.01 to QES-4B.10	QES-74D.01 to QES-74D.10	QES-75K.01 to QES-75K.10
QES-6.01 to QES-6.10	QES-74E.01 to QES-74E.10	QES-75Q.01 to QES-75Q.10
QES-23.01 to QES-23.10	QES-74K.01 to QES-74K.10	QES-75S.01 to QES-75S.10
QES-23A.01 to QES-23A.10	QES-74M.01 to QES-74M.10	QES-75V.01 to QES-75V.10
QES-28.01 to QES-28.10	QES-74Q.01 to QES-74Q.10	QES-75W.01 to QES-75W.10
QES-51.01 to QES-51.10	QES-74R.01 to QES-74R.10	QES-75Y.01 to QES-75Y.10
QES-52A.01 to QES-52A.10	QES-74V.01 to QES-74V.10	QES-75Z.01 to QES-75Z.10
QES-52D.01 to QES-52D.10	QES-75B.01 to QES-75B.10	QES-76F.01 to QES-76F.10
QES-55Dc.01 to QES-55Dc.10	QES-75D.01 to QES-75D.10	QES-76H.01 to QES-76H.10
QES-56A.01 to QES-56A.10	QES-75G.01 to QES-75G.10	QES-76K.01 to QES-76K.10
QES-56B.01 to QES-56B.10	QES-75H.01 to QES-75H.10	QES-76L.01 to QES-76L.10
QES-56C.01 to QES-56C.10	QES-75J.01 to QES-75J.10	QES-88B.01 to QES-88B.10
QES-56Ka.01 to QES-56Ka.10	EMPLOYER_ID.01 to EMPLOYER_ID.10	
QES-56Kb.01 to QES-56Kb.10	EMPLOYER_STARTDATE.01-M to EMPLOYER_STARTDATE.10-M	
QES-56Kc.01 to QES-56Kc.10	EMPLOYER_STARTDATE.01-D to EMPLOYER_STARTDATE.10-D	
QES-71A.01 to QES-71A.10	EMPLOYER_STARTDATE.01-Y to EMPLOYER_STARTDATE.10-Y	
QES-71I.01 to QES-71I.10	EMPLOYER_STOPDATE.01-M to EMPLOYER_STOPDATE.10-M	
QES-71J.01 to QES-71J.10	EMPLOYER_STOPDATE.01-D to EMPLOYER_STOPDATE.10-D	
QES-71P.01 to QES-71P.10	EMPLOYER_STOPDATE.01-Y to EMPLOYER_STOPDATE.10-Y	
QES-71R.01 to QES-71R.10	EMPLOYER_OCCODE.01 to EMPLOYER_OCCODE.10	
QES-73J.01 to QES-73J.10	EMPLOYER_INDCODE.01 to EMPLOYER_INDCODE.10	
	EMPLOYER_COWCODE.01 to EMPLOYER_COWCODE.10	

These variables repeat for employers AND gaps 1-4:

QES-30.01.01-M, ~D, ~Y to QES-30.01.04-M, ~D, ~Y
 QES-30.02.01-M, ~D, ~Y to QES-30.02.04-M, ~D, ~Y and so on through
 QES-30.10.01-M, ~D, ~Y to QES-30.10.04-M, ~D, ~Y
 QES-31.01.01-M, ~D, ~Y to QES-31.01.04-M, ~D, ~Y
 QES-31.02.01-M, ~D, ~Y to QES-31.02.04-M, ~D, ~Y and so on through
 QES-31.10.01-M, ~D, ~Y to QES-31.10.04-M, ~D, ~Y
 QES-33.01.01 to QES-33.01.04 through QES-33.10.01 to QES-33.10.04
 QES-34.01.01 to QES-34.01.04 through QES-34.10.01 to QES-34.10.04
 QES-36.01.01 to QES-36.01.04 through QES-36.10.01 to QES-36.10.04
 QES-40.01.01 to QES-40.01.04 through QES-40.10.01 to QES-40.10.04

2000 Work History Input Variables

/* The following variables were loaded into a relational data base and used as direct input for creation */
/* of the 1979–2000 work history data file.*/

PUBLIC_ID	These variables repeat for gaps 1-6:	
SAMPWEIGHT	Q7-10.01-D to Q7-10.06-D	
LINTDATE	Q7-10.01-M to Q7-10.06-M	
CURDATE	Q7-10.01-Y to Q7-10.06-Y	
Q4-1A, 1B	Q7-11.01-D to Q7-11.06-D	
Q4-5A	Q7-11.01-M to Q7-11.06-M	
Q4-6A-D, -M, -Y	Q7-11.01-Y to Q7-11.06-Y	
Q4-9, 9A	Q7-12.01 to Q7-12.06	
Q4-10	Q7-15.01 to Q7-15.06	
Q4-11, Q4-11B-D, -M, -Y	Q7-16.01 to Q7-16.06	
Q4-30	Q7-17.01 to Q7-17.06	
SYMBOL!ACTIVEFLAG	Q7-19.01 to Q7-19.06	

These variables repeat for employers 1-10:

QES-4B.01 to QES-4B.10	QES-73J.01 to QES-73J.10	QES-75J.01 to QES-75J.10
QES-6.01 to QES-6.10	QES-74D.01 to QES-74D.10	QES-75K.01 to QES-75K.10
QES-23.01 to QES-23.10	QES-74E.01 to QES-74E.10	QES-75Q.01 to QES-75Q.10
QES-23A.01 to QES-23A.10	QES-74K.01 to QES-74K.10	QES-75S.01 to QES-75S.10
QES-28.01 to QES-28.10	QES-74M.01 to QES-74M.10	QES-75V.01 to QES-75V.10
QES-51.01 to QES-51.10	QES-74Q.01 to QES-74Q.10	QES-75W.01 to QES-75W.10
QES-52A.01 to QES-52A.10	QES-74R.01 to QES-74R.10	QES-75Y.01 to QES-75Y.10
QES-52D.01 to QES-52D.10	QES-74U.01 to QES-74U.10	QES-75Z.01 to QES-75Z.10
QES-53B.01 to QES-53B.10	QES-74V.01 to QES-74V.10	QES-76F.01 to QES-76F.10
QES-54B.01 to QES-54B.10	QES-75B.01 to QES-75B.10	QES-76H.01 to QES-76H.10
QES-55Dc.01 to QES-55Dc.10	QES-75D.01 to QES-75D.10	QES-76K.01 to QES-76K.10
QES-56A.01 to QES-56A.10	QES-75G.01 to QES-75G.10	QES-76L.01 to QES-76L.10
QES-56B.01 to QES-56B.10	QES-75H.01 to QES-75H.10	QES-88B.01 to QES-88B.10
QES-56C.01 to QES-56C.10	EMPLOYER_ID.01 to EMPLOYER_ID.10	
QES-56Ka.01 to QES-56Ka.10	EMPLOYER_STARTDATE.01-M to EMPLOYER_STARTDATE.10-M	
QES-56Kb.01 to QES-56Kb.10	EMPLOYER_STARTDATE.01-D to EMPLOYER_STARTDATE.10-D	
QES-56Kc.01 to QES-56Kc.10	EMPLOYER_STARTDATE.01-Y to EMPLOYER_STARTDATE.10-Y	
QES-71A.01 to QES-71A.10	EMPLOYER_STOPDATE.01-M to EMPLOYER_STOPDATE.10-M	
QES-71I.01 to QES-71I.10	EMPLOYER_STOPDATE.01-D to EMPLOYER_STOPDATE.10-D	
QES-71J.01 to QES-71J.10	EMPLOYER_STOPDATE.01-Y to EMPLOYER_STOPDATE.10-Y	
QES-71P.01 to QES-71P.10	EMPLOYER_OCCODE.01 to EMPLOYER_OCCODE.10	
QES-71R.01 to QES-71R.10	EMPLOYER_INDCODE.01 to EMPLOYER_INDCODE.10	
	EMPLOYER_COWCODE.01 to EMPLOYER_COWCODE.10	

These variables repeat for employers AND gaps 1-4:

QES-30.01.01-M, -D, -Y to QES-30.01.04-M, -D, -Y
 QES-30.02.01-M, -D, -Y to QES-30.02.04-M, -D, -Y and so on through
 QES-30.10.01-M, -D, -Y to QES-30.10.04-M, -D, -Y
 QES-31.01.01-M, -D, -Y to QES-31.01.04-M, -D, -Y
 QES-31.02.01-M, -D, -Y to QES-31.02.04-M, -D, -Y and so on through
 QES-31.10.01-M, -D, -Y to QES-31.10.04-M, -D, -Y
 QES-33.01.01 to QES-33.01.04 through QES-33.10.01 to QES-33.10.04
 QES-34.01.01 to QES-34.01.04 through QES-34.10.01 to QES-34.10.04
 QES-36.01.01 to QES-36.01.04 through QES-36.10.01 to QES-36.10.04
 QES-40.01.01 to QES-40.01.04 through QES-40.10.01 to QES-40.10.04

NLSY79 WEEK NUMBERS AND CORRESPONDING DATES

The following list contains the start date for each week (Sunday) from January 1, 1978, through December 31, 2000, and the week numbers assigned to that week in the construction of the work history data file. These week numbers do not match the week numbers printed on the employment calendar included with the survey instrument materials. Week numbers assigned in the work history programs are assigned based upon actual dates collected during the course of the interview.

The variable names for the week-by-week arrays (status, hours, dual jobs) incorporate the specific year and number of the week within the specific year. For example, the 10th week in 1989 in the status array is called STAT8910. These names do not correspond to the strictly consecutive week numbers from 1–1201 listed below. The list below also contains the week numbers for each calendar year so that users will have a crosswalk for both calendar-year and continuous week numbers.

Week start date (Sunday)	Continuous week number	Calendar year week number
01-01-1978	1	1
01-08-1978	2	2
01-15-1978	3	3
01-22-1978	4	4
01-29-1978	5	5
02-05-1978	6	6
02-12-1978	7	7
02-19-1978	8	8
02-26-1978	9	9
03-05-1978	10	10
03-12-1978	11	11
03-19-1978	12	12
03-26-1978	13	13
04-02-1978	14	14
04-09-1978	15	15
04-16-1978	16	16
04-23-1978	17	17
04-30-1978	18	18
05-07-1978	19	19
05-14-1978	20	20
05-21-1978	21	21
05-28-1978	22	22
06-04-1978	23	23
06-11-1978	24	24
06-18-1978	25	25
06-25-1978	26	26
07-02-1978	27	27
07-09-1978	28	28
07-16-1978	29	29
07-23-1978	30	30
07-30-1978	31	31

Week start date (Sunday)	Continuous week number	Calendar year week number
08-06-1978	32	32
08-13-1978	33	33
08-20-1978	34	34
08-27-1978	35	35
09-03-1978	36	36
09-10-1978	37	37
09-17-1978	38	38
09-24-1978	39	39
10-01-1978	40	40
10-08-1978	41	41
10-15-1978	42	42
10-22-1978	43	43
10-29-1978	44	44
11-05-1978	45	45
11-12-1978	46	46
11-19-1978	47	47
11-26-1978	48	48
12-03-1978	49	49
12-10-1978	50	50
12-17-1978	51	51
12-24-1978	52	52
12-31-1978	53	53
01-07-1979	54	1
01-14-1979	55	2
01-21-1979	56	3
01-28-1979	57	4
02-04-1979	58	5
02-11-1979	59	6
02-18-1979	60	7
02-25-1979	61	8
03-04-1979	62	9

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
03-11-1979	63	10	01-13-1980	107	2
03-18-1979	64	11	01-20-1980	108	3
03-25-1979	65	12	01-27-1980	109	4
04-01-1979	66	13	02-03-1980	110	5
04-08-1979	67	14	02-10-1980	111	6
04-15-1979	68	15	02-17-1980	112	7
04-22-1979	69	16	02-24-1980	113	8
04-29-1979	70	17	03-02-1980	114	9
05-06-1979	71	18	03-09-1980	115	10
05-13-1979	72	19	03-16-1980	116	11
05-20-1979	73	20	03-23-1980	117	12
05-27-1979	74	21	03-30-1980	118	13
06-03-1979	75	22	04-06-1980	119	14
06-10-1979	76	23	04-13-1980	120	15
06-17-1979	77	24	04-20-1980	121	16
06-24-1979	78	25	04-27-1980	122	17
07-01-1979	79	26	05-04-1980	123	18
07-08-1979	80	27	05-11-1980	124	19
07-15-1979	81	28	05-18-1980	125	20
07-22-1979	82	29	05-25-1980	126	21
07-29-1979	83	30	06-01-1980	127	22
08-05-1979	84	31	06-08-1980	128	23
08-12-1979	85	32	06-15-1980	129	24
08-19-1979	86	33	06-22-1980	130	25
08-26-1979	87	34	06-29-1980	131	26
09-02-1979	88	35	07-06-1980	132	27
09-09-1979	89	36	07-13-1980	133	28
09-16-1979	90	37	07-20-1980	134	29
09-23-1979	91	38	07-27-1980	135	30
09-30-1979	92	39	08-03-1980	136	31
10-07-1979	93	40	08-10-1980	137	32
10-14-1979	94	41	08-17-1980	138	33
10-21-1979	95	42	08-24-1980	139	34
10-28-1979	96	43	08-31-1980	140	35
11-04-1979	97	44	09-07-1980	141	36
11-11-1979	98	45	09-14-1980	142	37
11-18-1979	99	46	09-21-1980	143	38
11-25-1979	100	47	09-28-1980	144	39
12-02-1979	101	48	10-05-1980	145	40
12-09-1979	102	49	10-12-1980	146	41
12-16-1979	103	50	10-19-1980	147	42
12-23-1979	104	51	10-26-1980	148	43
12-30-1979	105	52	11-02-1980	149	44
01-06-1980	106	1	11-09-1980	150	45

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number
11-16-1980	151	46
11-23-1980	152	47
11-30-1980	153	48
12-07-1980	154	49
12-14-1980	155	50
12-21-1980	156	51
12-28-1980	157	52
01-04-1981	158	1
01-11-1981	159	2
01-18-1981	160	3
01-25-1981	161	4
02-01-1981	162	5
02-08-1981	163	6
02-15-1981	164	7
02-22-1981	165	8
03-01-1981	166	9
03-08-1981	167	10
03-15-1981	168	11
03-22-1981	169	12
03-29-1981	170	13
04-05-1981	171	14
04-12-1981	172	15
04-19-1981	173	16
04-26-1981	174	17
05-03-1981	175	18
05-10-1981	176	19
05-17-1981	177	20
05-24-1981	178	21
05-31-1981	179	22
06-07-1981	180	23
06-14-1981	181	24
06-21-1981	182	25
06-28-1981	183	26
07-05-1981	184	27
07-12-1981	185	28
07-19-1981	186	29
07-26-1981	187	30
08-02-1981	188	31
08-09-1981	189	32
08-16-1981	190	33
08-23-1981	191	34
08-30-1981	192	35
09-06-1981	193	36
09-13-1981	194	37

Week start date (Sunday)	Continuous week number	Calendar year week number
09-20-1981	195	38
09-27-1981	196	39
10-04-1981	197	40
10-11-1981	198	41
10-18-1981	199	42
10-25-1981	200	43
11-01-1981	201	44
11-08-1981	202	45
11-15-1981	203	46
11-22-1981	204	47
11-29-1981	205	48
12-06-1981	206	49
12-13-1981	207	50
12-20-1981	208	51
12-27-1981	209	52
01-03-1982	210	1
01-10-1982	211	2
01-17-1982	212	3
01-24-1982	213	4
01-31-1982	214	5
02-07-1982	215	6
02-14-1982	216	7
02-21-1982	217	8
02-28-1982	218	9
03-07-1982	219	10
03-14-1982	220	11
03-21-1982	221	12
03-28-1982	222	13
04-04-1982	223	14
04-11-1982	224	15
04-18-1982	225	16
04-25-1982	226	17
05-02-1982	227	18
05-09-1982	228	19
05-16-1982	229	20
05-23-1982	230	21
05-30-1982	231	22
06-06-1982	232	23
06-13-1982	233	24
06-20-1982	234	25
06-27-1982	235	26
07-04-1982	236	27
07-11-1982	237	28
07-18-1982	238	29

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
07-25-1982	239	30	05-29-1983	283	22
08-01-1982	240	31	06-05-1983	284	23
08-08-1982	241	32	06-12-1983	285	24
08-15-1982	242	33	06-19-1983	286	25
08-22-1982	243	34	06-26-1983	287	26
08-29-1982	244	35	07-03-1983	288	27
09-05-1982	245	36	07-10-1983	289	28
09-12-1982	246	37	07-17-1983	290	29
09-29-1982	247	38	07-24-1983	291	30
09-26-1982	248	39	07-31-1983	292	31
10-03-1982	249	40	08-07-1983	293	32
10-10-1982	250	41	08-14-1983	294	33
10-17-1982	251	42	08-21-1983	295	34
10-24-1982	252	43	08-28-1983	296	35
10-31-1982	253	44	09-04-1983	297	36
11-07-1982	254	45	09-11-1983	298	37
11-14-1982	255	46	09-18-1983	299	38
11-21-1982	256	47	09-25-1983	300	39
11-28-1982	257	48	10-02-1983	301	40
12-05-1982	258	49	10-09-1983	302	41
12-12-1982	259	50	10-16-1983	303	42
12-19-1982	260	51	10-23-1983	304	43
12-26-1982	261	52	10-30-1983	305	44
01-02-1983	262	1	11-06-1983	306	45
01-09-1983	263	2	11-13-1983	307	46
01-16-1983	264	3	11-20-1983	308	47
01-23-1983	265	4	11-27-1983	309	48
01-30-1983	266	5	12-04-1983	310	49
02-06-1983	267	6	12-11-1983	311	50
02-13-1983	268	7	12-18-1983	312	51
02-20-1983	269	8	12-25-1983	313	52
02-27-1983	270	9	01-01-1984	314	1
03-06-1983	271	10	01-08-1984	315	2
03-13-1983	272	11	01-15-1984	316	3
03-20-1983	273	12	01-22-1984	317	4
03-27-1983	274	13	01-29-1984	318	5
04-03-1983	275	14	02-05-1984	319	6
04-10-1983	276	15	02-12-1984	320	7
04-17-1983	277	16	02-19-1984	321	8
04-24-1983	278	17	02-26-1984	322	9
05-01-1983	279	18	03-04-1984	323	10
05-08-1983	280	19	03-11-1984	324	11
05-15-1983	281	20	03-18-1984	325	12
05-22-1983	282	21	03-25-1984	326	13

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number
04-01-1984	327	14
04-08-1984	328	15
04-15-1984	329	16
04-22-1984	330	17
04-29-1984	331	18
05-06-1984	332	19
05-13-1984	333	20
05-20-1984	334	21
05-27-1984	335	22
06-03-1984	336	23
06-10-1984	337	24
06-17-1984	338	25
06-24-1984	339	26
07-01-1984	340	27
07-08-1984	341	28
07-15-1984	342	29
07-22-1984	343	30
07-29-1984	344	31
08-05-1984	345	32
08-12-1984	346	33
08-19-1984	347	34
08-26-1984	348	35
09-02-1984	349	36
09-09-1984	350	37
09-16-1984	351	38
09-23-1984	352	39
09-30-1984	353	40
10-07-1984	354	41
10-14-1984	355	42
10-21-1984	356	43
10-28-1984	357	44
11-04-1984	358	45
11-11-1984	359	46
11-18-1984	360	47
11-25-1984	361	48
12-02-1984	362	49
12-09-1984	363	50
12-16-1984	364	51
12-23-1984	365	52
12-30-1984	366	53
01-06-1985	367	1
01-13-1985	368	2
01-20-1985	369	3
01-27-1985	370	4

Week start date (Sunday)	Continuous week number	Calendar year week number
02-03-1985	371	5
02-10-1985	372	6
02-17-1985	373	7
02-24-1985	374	8
03-03-1985	375	9
03-10-1985	376	10
03-17-1985	377	11
03-24-1985	378	12
03-31-1985	379	13
04-07-1985	380	14
04-14-1985	381	15
04-21-1985	382	16
04-28-1985	383	17
05-05-1985	384	18
05-12-1985	385	19
05-19-1985	386	20
05-26-1985	387	21
06-02-1985	388	22
06-09-1985	389	23
06-16-1985	390	24
06-23-1985	391	25
06-30-1985	392	26
07-07-1985	393	27
07-14-1985	394	28
07-21-1985	395	29
07-28-1985	396	30
08-04-1985	397	31
08-11-1985	398	32
08-18-1985	399	33
08-25-1985	400	34
09-01-1985	401	35
09-08-1985	402	36
09-15-1985	403	37
09-22-1985	404	38
09-29-1985	405	39
10-06-1985	406	40
10-13-1985	407	41
10-20-1985	408	42
10-27-1985	409	43
11-03-1985	410	44
11-10-1985	411	45
11-17-1985	412	46
11-24-1985	413	47
12-01-1985	414	48

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
12-08-1985	415	49	10-12-1986	459	41
12-15-1985	416	50	10-19-1986	460	42
12-22-1985	417	51	10-26-1986	461	43
12-29-1985	418	52	11-02-1986	462	44
01-05-1986	419	1	11-09-1986	463	45
01-12-1986	420	2	11-16-1986	464	46
01-19-1986	421	3	11-23-1986	465	47
01-26-1986	422	4	11-30-1986	466	48
02-02-1986	423	5	12-07-1986	467	49
02-09-1986	424	6	12-14-1986	468	50
02-16-1986	425	7	12-21-1986	469	51
02-23-1986	426	8	12-28-1986	470	52
03-02-1986	427	9	01-04-1987	471	1
03-09-1986	428	10	01-11-1987	472	2
03-16-1986	429	11	01-18-1987	473	3
03-23-1986	430	12	01-25-1987	474	4
03-30-1986	431	13	02-01-1987	475	5
04-06-1986	432	14	02-08-1987	476	6
04-13-1986	433	15	02-15-1987	477	7
04-20-1986	434	16	02-22-1987	478	8
04-27-1986	435	17	03-01-1987	479	9
05-04-1986	436	18	03-08-1987	480	10
05-11-1986	437	19	03-15-1987	481	11
05-18-1986	438	20	03-22-1987	482	12
05-25-1986	439	21	03-29-1987	483	13
06-01-1986	440	22	04-05-1987	484	14
06-08-1986	441	23	04-12-1987	485	15
06-15-1986	442	24	04-19-1987	486	16
06-22-1986	443	25	04-26-1987	487	17
06-29-1986	444	26	05-03-1987	488	18
07-06-1986	445	27	05-10-1987	489	19
07-13-1986	446	28	05-17-1987	490	20
07-20-1986	447	29	05-24-1987	491	21
07-27-1986	448	30	05-31-1987	492	22
08-03-1986	449	31	06-07-1987	493	23
08-10-1986	450	32	06-14-1987	494	24
08-17-1986	451	33	06-21-1987	495	25
08-24-1986	452	34	06-28-1987	496	26
08-31-1986	453	35	07-05-1987	497	27
09-07-1986	454	36	07-12-1987	498	28
09-14-1986	455	37	07-19-1987	499	29
09-21-1986	456	38	07-26-1987	500	30
09-28-1986	457	39	08-02-1987	501	31
10-05-1986	458	40	08-09-1987	502	32

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number
08-16-1987	503	33
08-23-1987	504	34
08-30-1987	505	35
09-06-1987	506	36
09-13-1987	507	37
09-20-1987	508	38
09-27-1987	509	39
10-04-1987	510	40
10-11-1987	511	41
10-18-1987	512	42
10-25-1987	513	43
11-01-1987	514	44
11-08-1987	515	45
11-15-1987	516	46
11-22-1987	517	47
11-29-1987	518	48
12-06-1987	519	49
12-13-1987	520	50
12-20-1987	521	51
12-27-1987	522	52
01-03-1988	523	1
01-10-1988	524	2
01-17-1988	525	3
01-24-1988	526	4
01-31-1988	527	5
02-07-1988	528	6
02-14-1988	529	7
02-21-1988	530	8
02-28-1988	531	9
03-06-1988	532	10
03-13-1988	533	11
03-20-1988	534	12
03-27-1988	535	13
04-03-1988	536	14
04-10-1988	537	15
04-17-1988	538	16
04-24-1988	539	17
05-01-1988	540	18
05-08-1988	541	19
05-15-1988	542	20
05-22-1988	543	21
05-29-1988	544	22
06-05-1988	545	23
06-12-1988	546	24

Week start date (Sunday)	Continuous week number	Calendar year week number
06-19-1988	547	25
06-26-1988	548	26
07-03-1988	549	27
07-10-1988	550	28
07-17-1988	551	29
07-24-1988	552	30
07-31-1988	553	31
08-07-1988	554	32
08-14-1988	555	33
08-21-1988	556	34
08-28-1988	557	35
09-04-1988	558	36
09-11-1988	559	37
09-18-1988	560	38
09-25-1988	561	39
10-02-1988	562	40
10-09-1988	563	41
10-16-1988	564	42
10-23-1988	565	43
10-30-1988	566	44
11-06-1988	567	45
11-13-1988	568	46
11-20-1988	569	47
11-27-1988	570	48
12-04-1988	571	49
12-11-1988	572	50
12-18-1988	573	51
12-25-1988	574	52
01-01-1989	575	1
01-08-1989	576	2
01-15-1989	577	3
01-22-1989	578	4
01-29-1989	579	5
02-05-1989	580	6
02-12-1989	581	7
02-19-1989	582	8
02-26-1989	583	9
03-05-1989	584	10
03-12-1989	585	11
03-19-1989	586	12
03-26-1989	587	13
04-02-1989	588	14
04-09-1989	589	15
04-16-1989	590	16

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
04-23-1989	591	17	02-25-1990	635	8
04-30-1989	592	18	03-04-1990	636	9
05-07-1989	593	19	03-11-1990	637	10
05-14-1989	594	20	03-18-1990	638	11
05-21-1989	595	21	03-25-1990	639	12
05-28-1989	596	22	04-01-1990	640	13
06-04-1989	597	23	04-08-1990	641	14
06-11-1989	598	24	04-15-1990	642	15
06-18-1989	599	25	04-22-1990	643	16
06-25-1989	600	26	04-29-1990	644	17
07-02-1989	601	27	05-06-1990	645	18
07-09-1989	602	28	05-13-1990	646	19
07-16-1989	603	29	05-20-1990	647	20
07-23-1989	604	30	05-27-1990	648	21
07-30-1989	605	31	06-03-1990	649	22
08-06-1989	606	32	06-10-1990	650	23
08-13-1989	607	33	06-17-1990	651	24
08-20-1989	608	34	06-24-1990	652	25
08-27-1989	609	35	07-01-1990	653	26
09-03-1989	610	36	07-08-1990	654	27
09-10-1989	611	37	07-15-1990	655	28
09-17-1989	612	38	07-22-1990	656	29
09-24-1989	613	39	07-29-1990	657	30
10-01-1989	614	40	08-05-1990	658	31
10-08-1989	615	41	08-12-1990	659	32
10-15-1989	616	42	08-19-1990	660	33
10-22-1989	617	43	08-26-1990	661	34
10-29-1989	618	44	09-02-1990	662	35
11-05-1989	619	45	09-09-1990	663	36
11-12-1989	620	46	09-16-1990	664	37
11-19-1989	621	47	09-23-1990	665	38
11-26-1989	622	48	09-30-1990	666	39
12-03-1989	623	49	10-07-1990	667	40
12-10-1989	624	50	10-14-1990	668	41
12-17-1989	625	51	10-21-1990	669	42
12-24-1989	626	52	10-28-1990	670	43
12-31-1989	627	53	11-04-1990	671	44
01-07-1990	628	1	11-11-1990	672	45
01-14-1990	629	2	11-18-1990	673	46
01-21-1990	630	3	11-25-1990	674	47
01-28-1990	631	4	12-02-1990	675	48
02-04-1990	632	5	12-09-1990	676	49
02-11-1990	633	6	12-16-1990	677	50
02-18-1990	634	7	12-23-1990	678	51

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number
12-30-1990	679	52
01-06-1991	680	1
01-13-1991	681	2
01-20-1991	682	3
01-27-1991	683	4
02-03-1991	684	5
02-10-1991	685	6
02-17-1991	686	7
02-24-1991	687	8
03-03-1991	688	9
03-10-1991	689	10
03-17-1991	690	11
03-24-1991	691	12
03-31-1991	692	13
04-07-1991	693	14
04-14-1991	694	15
04-21-1991	695	16
04-28-1991	696	17
05-05-1991	697	18
05-12-1991	698	19
05-19-1991	699	20
05-26-1991	700	21
06-02-1991	701	22
06-09-1991	702	23
06-16-1991	703	24
06-23-1991	704	25
06-30-1991	705	26
07-07-1991	706	27
07-14-1991	707	28
07-21-1991	708	29
07-28-1991	709	30
08-04-1991	710	31
08-11-1991	711	32
08-18-1991	712	33
08-25-1991	713	34
09-01-1991	714	35
09-08-1991	715	36
09-15-1991	716	37
09-22-1991	717	38
09-29-1991	718	39
10-06-1991	719	40
10-13-1991	720	41
10-20-1991	721	42
10-27-1991	722	43

Week start date (Sunday)	Continuous week number	Calendar year week number
11-03-1991	723	44
11-10-1991	724	45
11-17-1991	725	46
11-24-1991	726	47
12-01-1991	727	48
12-08-1991	728	49
12-15-1991	729	50
12-22-1991	730	51
12-29-1991	731	52
01-05-1992	732	1
01-12-1992	733	2
01-19-1992	734	3
01-26-1992	735	4
02-02-1992	736	5
02-09-1992	737	6
02-16-1992	738	7
02-23-1992	739	8
03-01-1992	740	9
03-08-1992	741	10
03-15-1992	742	11
03-22-1992	743	12
03-29-1992	744	13
04-05-1992	745	14
04-12-1992	746	15
04-19-1992	747	16
04-26-1992	748	17
05-03-1992	749	18
05-10-1992	750	19
05-17-1992	751	20
05-24-1992	752	21
05-31-1992	753	22
06-07-1992	754	23
06-14-1992	755	24
06-21-1992	756	25
06-28-1992	757	26
07-05-1992	758	27
07-12-1992	759	28
07-19-1992	760	29
07-26-1992	761	30
08-02-1992	762	31
08-09-1992	763	32
08-16-1992	764	33
08-23-1992	765	34
08-30-1992	766	35

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
09-06-1992	767	36	07-11-1993	811	28
09-13-1992	768	37	07-18-1993	812	29
09-20-1992	769	38	07-25-1993	813	30
09-27-1992	770	39	08-01-1993	814	31
10-04-1992	771	40	08-08-1993	815	32
10-11-1992	772	41	08-15-1993	816	33
10-18-1992	773	42	08-22-1993	817	34
10-25-1992	774	43	08-29-1993	818	35
11-01-1992	775	44	09-05-1993	819	36
11-08-1992	776	45	09-12-1993	820	37
11-15-1992	777	46	09-19-1993	821	38
11-22-1992	778	47	09-26-1993	822	39
11-29-1992	779	48	10-03-1993	823	40
12-06-1992	780	49	10-10-1993	824	41
12-13-1992	781	50	10-17-1993	825	42
12-20-1992	782	51	10-24-1993	826	43
12-27-1992	783	52	10-31-1993	827	44
01-03-1993	784	1	11-07-1993	828	45
01-10-1993	785	2	11-14-1993	829	46
01-17-1993	786	3	11-21-1993	830	47
01-24-1993	787	4	11-28-1993	831	48
01-31-1993	788	5	12-05-1993	832	49
02-07-1993	789	6	12-12-1993	833	50
02-14-1993	790	7	12-19-1993	834	51
02-21-1993	791	8	12-26-1993	835	52
02-28-1993	792	9	01-02-1994	836	1
03-07-1993	793	10	01-09-1994	837	2
03-14-1993	794	11	01-16-1994	838	3
03-21-1993	795	12	01-23-1994	839	4
03-28-1993	796	13	01-30-1994	840	5
04-04-1993	797	14	02-06-1994	841	6
04-11-1993	798	15	02-13-1994	842	7
04-18-1993	799	16	02-20-1994	843	8
04-25-1993	800	17	02-27-1994	844	9
05-02-1993	801	18	03-06-1994	845	10
05-09-1993	802	19	03-13-1994	846	11
05-16-1993	803	20	03-20-1994	847	12
05-23-1993	804	21	03-27-1994	848	13
05-30-1993	805	22	04-03-1994	849	14
06-06-1993	806	23	04-10-1994	850	15
06-13-1993	807	24	04-17-1994	851	16
06-20-1993	808	25	04-24-1994	852	17
06-27-1993	809	26	05-01-1994	853	18
07-04-1993	810	27	05-08-1994	854	19

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number
05-15-1994	855	20
05-22-1994	856	21
05-29-1994	857	22
06-05-1994	858	23
06-12-1994	859	24
06-19-1994	860	25
06-26-1994	861	26
07-03-1994	862	27
07-10-1994	863	28
07-17-1994	864	29
07-24-1994	865	30
07-31-1994	866	31
08-07-1994	867	32
08-14-1994	868	33
08-21-1994	869	34
08-28-1994	870	35
09-04-1994	871	36
09-11-1994	872	37
09-18-1994	873	38
09-25-1994	874	39
10-02-1994	875	40
10-09-1994	876	41
10-16-1994	877	42
10-23-1994	878	43
10-30-1994	879	44
11-06-1994	880	45
11-13-1994	881	46
11-20-1994	882	47
11-27-1994	883	48
12-04-1994	884	49
12-11-1994	885	50
12-18-1994	886	51
12-25-1994	887	52
01-01-1995	888	1
01-08-1995	889	2
01-15-1995	890	3
01-22-1995	891	4
01-29-1995	892	5
02-05-1995	893	6
02-12-1995	894	7
02-19-1995	895	8
02-26-1995	896	9
03-05-1995	897	10
03-12-1995	898	11

Week start date (Sunday)	Continuous week number	Calendar year week number
03-19-1995	899	12
03-26-1995	900	13
04-02-1995	901	14
04-09-1995	902	15
04-16-1995	903	16
04-23-1995	904	17
04-30-1995	905	18
05-07-1995	906	19
05-14-1995	907	20
05-21-1995	908	21
05-28-1995	909	22
06-04-1995	910	23
06-11-1995	911	24
06-18-1995	912	25
06-25-1995	913	26
07-02-1995	914	27
07-09-1995	915	28
07-16-1995	916	29
07-23-1995	917	30
07-30-1995	918	31
08-06-1995	919	32
08-13-1995	920	33
08-20-1995	921	34
08-27-1995	922	35
09-03-1995	923	36
09-10-1995	924	37
09-17-1995	925	38
09-24-1995	926	39
10-01-1995	927	40
10-08-1995	928	41
10-15-1995	929	42
10-22-1995	930	43
10-29-1995	931	44
11-05-1995	932	45
11-12-1995	933	46
11-19-1995	934	47
11-26-1995	935	48
12-03-1995	936	49
12-10-1995	937	50
12-17-1995	938	51
12-24-1995	939	52
12-31-1995	940	53
01-07-1996	941	1
01-14-1996	942	2

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
01-21-1996	943	3	11-24-1996	987	47
01-28-1996	944	4	12-01-1996	988	48
02-04-1996	945	5	12-08-1996	989	49
02-11-1996	946	6	12-15-1996	990	50
02-18-1996	947	7	12-22-1996	991	51
02-25-1996	948	8	12-29-1996	992	52
03-03-1996	949	9	01-05-1997	993	1
03-10-1996	950	10	01-12-1997	994	2
03-17-1996	951	11	01-19-1997	995	3
03-24-1996	952	12	01-26-1997	996	4
03-31-1996	953	13	02-02-1997	997	5
04-07-1996	954	14	02-09-1997	998	6
04-14-1996	955	15	02-16-1997	999	7
04-21-1996	956	16	02-23-1997	1000	8
04-28-1996	957	17	03-02-1997	1001	9
05-05-1996	958	18	03-09-1997	1002	10
05-12-1996	959	19	03-16-1997	1003	11
05-19-1996	960	20	03-23-1997	1004	12
05-26-1996	961	21	03-30-1997	1005	13
06-02-1996	962	22	04-06-1997	1006	14
06-09-1996	963	23	04-13-1997	1007	15
06-16-1996	964	24	04-20-1997	1008	16
06-23-1996	965	25	04-27-1997	1009	17
06-30-1996	966	26	05-04-1997	1010	18
07-07-1996	967	27	05-11-1997	1011	19
07-14-1996	968	28	05-18-1997	1012	20
07-21-1996	969	29	05-25-1997	1013	21
07-28-1996	970	30	06-01-1997	1014	22
08-04-1996	971	31	06-08-1997	1015	23
08-11-1996	972	32	06-15-1997	1016	24
08-18-1996	973	33	06-22-1997	1017	25
08-25-1996	974	34	06-29-1997	1018	26
09-01-1996	975	35	07-06-1997	1019	27
09-08-1996	976	36	07-13-1997	1020	28
09-15-1996	977	37	07-20-1997	1021	29
09-22-1996	978	38	07-27-1997	1022	30
09-29-1996	979	39	08-03-1997	1023	31
10-06-1996	980	40	08-10-1997	1024	32
10-13-1996	981	41	08-17-1997	1025	33
10-20-1996	982	42	08-24-1997	1026	34
10-27-1996	983	43	08-31-1997	1027	35
11-03-1996	984	44	09-07-1997	1028	36
11-10-1996	985	45	09-14-1997	1029	37
11-17-1996	986	46	09-21-1997	1030	38

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number
09-28-1997	1031	39
10-05-1997	1032	40
10-12-1997	1033	41
10-19-1997	1034	42
10-26-1997	1035	43
11-02-1997	1036	44
11-09-1997	1037	45
11-16-1997	1038	46
11-23-1997	1039	47
11-30-1997	1040	48
12-07-1997	1041	49
12-14-1997	1042	50
12-21-1997	1043	51
12-28-1997	1044	52
01-04-1998	1045	1
01-11-1998	1046	2
01-18-1998	1047	3
01-25-1998	1048	4
02-01-1998	1049	5
02-08-1998	1050	6
02-15-1998	1051	7
02-22-1998	1052	8
03-01-1998	1053	9
03-08-1998	1054	10
03-15-1998	1055	11
03-22-1998	1056	12
03-29-1998	1057	13
04-05-1998	1058	14
04-12-1998	1059	15
04-19-1998	1060	16
04-26-1998	1061	17
05-03-1998	1062	18
05-10-1998	1063	19
05-17-1998	1064	20
05-24-1998	1065	21
05-31-1998	1066	22
06-07-1998	1067	23
06-14-1998	1068	24
06-21-1998	1069	25
06-28-1998	1070	26
07-05-1998	1071	27
07-12-1998	1072	28
07-19-1998	1073	29
07-26-1998	1074	30

Week start date (Sunday)	Continuous week number	Calendar year week number
08-02-1998	1075	31
08-09-1998	1076	32
08-16-1998	1077	33
08-23-1998	1078	34
08-30-1998	1079	35
09-06-1998	1080	36
09-13-1998	1081	37
09-20-1998	1082	38
09-27-1998	1083	39
10-04-1998	1084	40
10-11-1998	1085	41
10-18-1998	1086	42
10-25-1998	1087	43
11-01-1998	1088	44
11-08-1998	1089	45
11-15-1998	1090	46
11-22-1998	1091	47
11-29-1998	1092	48
12-06-1998	1093	49
12-13-1998	1094	50
12-20-1998	1095	51
12-27-1998	1096	52
01-03-1999	1097	1
01-10-1999	1098	2
01-17-1999	1099	3
01-24-1999	1100	4
01-31-1999	1101	5
02-07-1999	1102	6
02-14-1999	1103	7
02-21-1999	1104	8
02-28-1999	1105	9
03-07-1999	1106	10
03-14-1999	1107	11
03-21-1999	1108	12
03-28-1999	1109	13
04-04-1999	1110	14
04-11-1999	1111	15
04-18-1999	1112	16
04-25-1999	1113	17
05-02-1999	1114	18
05-09-1999	1115	19
05-16-1999	1116	20
05-23-1999	1117	21
05-30-1999	1118	22

Appendix 18: Work History Data

Week start date (Sunday)	Continuous week number	Calendar year week number	Week start date (Sunday)	Continuous week number	Calendar year week number
06-06-1999	1119	23	04-09-2000	1163	15
06-13-1999	1120	24	04-16-2000	1164	16
06-20-1999	1121	25	04-23-2000	1165	17
06-27-1999	1122	26	04-30-2000	1166	18
07-04-1999	1123	27	05-07-2000	1167	19
07-11-1999	1124	28	05-14-2000	1168	20
07-18-1999	1125	29	05-21-2000	1169	21
07-25-1999	1126	30	05-28-2000	1170	22
08-01-1999	1127	31	06-04-2000	1171	23
08-08-1999	1128	32	06-11-2000	1172	24
08-15-1999	1129	33	06-18-2000	1173	25
08-22-1999	1130	34	06-25-2000	1174	26
08-29-1999	1131	35	07-02-2000	1175	27
09-05-1999	1132	36	07-09-2000	1176	28
09-12-1999	1133	37	07-16-2000	1177	29
09-19-1999	1134	38	07-23-2000	1178	30
09-26-1999	1135	39	07-30-2000	1179	31
10-03-1999	1136	40	08-06-2000	1180	32
10-10-1999	1137	41	08-13-2000	1181	33
10-17-1999	1138	42	08-20-2000	1182	34
10-24-1999	1139	43	08-27-2000	1183	35
10-31-1999	1140	44	09-03-2000	1184	36
11-07-1999	1141	45	09-10-2000	1185	37
11-14-1999	1142	46	09-17-2000	1186	38
11-21-1999	1143	47	09-24-2000	1187	39
11-28-1999	1144	48	10-01-2000	1188	40
12-05-1999	1145	49	10-08-2000	1189	41
12-12-1999	1146	50	10-15-2000	1190	42
12-19-1999	1147	51	10-23-2000	1191	43
12-26-1999	1148	52	10-29-2000	1192	44
01-02-2000	1149	1	11-05-2000	1193	45
01-09-2000	1150	2	11-12-2000	1194	46
01-16-2000	1151	3	11-19-2000	1195	47
01-23-2000	1152	4	11-26-2000	1196	48
01-30-2000	1153	5	12-03-2000	1197	49
02-06-2000	1154	6	12-10-2000	1198	50
02-13-2000	1155	7	12-17-2000	1199	51
02-20-2000	1156	8	12-24-2000	1200	52
02-27-2000	1157	9	12-31-2000	1201	53
03-05-2000	1158	10			
03-12-2000	1159	11			
03-19-2000	1160	12			
03-26-2000	1161	13			
04-02-2000	1162	14			

NLSY79 APPENDIX 19:

SF-12 HEALTH SCALE SCORING

SF-12 Summary Scores

The SF-12, which stands for short-form 12-question, is a brief inventory of self-reported mental and physical health. This scale was administered to respondents who had turned 40 since their last interview as part of the age 40+ health module, included in the 1998 and 2000 surveys (and planned for inclusion in 2002 and 2004).

Rather than using the twelve questions separately, SF-12 users often create two summary scores:

- PCS-12 or Physical Component Summary (measures physical health): question name Q11-H40SF12_PCS_SCORE
- MCS-12 or Mental Component Summary (measures mental health): question name Q11-H40SF12_MCS_SCORE

CHRR has received permission to calculate these summary scores for NLSY79 respondents and release the scores with the main data set. Scores are created according to the manual by Ware, Kosinski, and Keller (1995) and are provided on the data set with the question names listed above. However, we are not permitted to release the scoring formula; interested users can purchase the scoring manual from Quality Metric (<http://www.qmetric.com/products/publications/sf12.php3>).

In large national surveys of the entire US population, both the PCS-12 and MCS-12 have a mean of 50 and a standard deviation of 10. The interpretation of these two scores is straightforward. NLSY79 respondents with a score above 50 have better health than the typical person in the general U.S. population (age is not held constant). NLSY79 respondents with scores below 50 have worse health than the typical U.S. person. Each one-point difference above or below 50 corresponds to a one-tenth of a standard deviation. For example, a person with a score of 30 is two standard deviations away from the mean.

TABLE 1: Summary Statistics for NLSY79 SF-12 Scores

	1998	2000
PCS	Mean: 52.54234 StDev: 7.36181	Mean: 52.52280 StDev: 7.77585
MCS	Mean: 53.09726 StDev: 7.84907	Mean: 53.05464 StDev: 8.25104

As Table 1 indicates, the typical NLSY79 respondent self-reports better health than the typical U.S. respondent. This matches the results described in the SF-12 scoring manual, which in addition to population norms reports the norms for U.S. residents who are between the ages of 35 and 44. The manual reports the mean PCS score for this subgroup as 52.18 (std. dev. 7.30) and the mean MCS score as 50.1 (std. dev. 8.62). The SF-12 manual reports higher-than-average physical scores prior to age 55 and rapidly falling scores after that age. Mental scores do not appear to decline with age.

Reference

Ware, John, Mark Kosinski and Susan Keller. 1995. *SF-12: How to Score the SF-12 Physical and Mental Health Summary Scales, 2nd edition*. Boston: The Health Institute, New England Medical Center.